Towards an architecture for the support of integrated personal health records

Bastianen, S.G.A.

Award date:
2015

Eindhoven, March 2015

Towards an architecture for the support of integrated Personal Health Records

by
S.G.A. Bastianen

BSc Industrial Engineering & Management Science
Student identity number 0656423

in partial fulfilment of the requirements for the degree of

Master of Science
in Business Information Systems

Supervisors

Dr. P.M.E. Van Gorp	Eindhoven University of Technology *Industrial Engineering and Innovation Sciences*
Dr. M. Pechenizkiy	Eindhoven University of Technology *Mathematics and Computer Science*
F. Ploegmakers	SAP Nederland

The nice thing about standards is that you have so many to choose from.

Andrew S. Tanenbaum

Abstract

In the past decade Care Delivery Organizations (CDOs) have widely adopted Electronic Medical Record (EMR) systems for storing patient records digitally. As information technology is developing and healthcare is becoming more patient-centered, we notice the development of Personal Health Record (PHR) systems, which enable individuals to store their health-related information in a digital, online fashion. Together with the upcoming trend of Wearable Health Monitoring Systems (WHMSs), which ideally store their information in a PHR system, it is expected that individuals have a valuable collection of health information that can support the healthcare services that they receive.
In this research we investigate the technical challenges and barriers for integrating EMR systems and PHR systems and how these can be resolved in order to come to a more holistic approach in the way health information is used. We do this by analyzing currently existing PHR systems and investigating the current health information infrastructure in The Netherlands, while having a strong focus on interoperability standards, e.g. HL7 Fast Healthcare Interoperability Resources (FHIR). In order to also cover challenges that arise from a practical sense, we have built a prototype of an integrated PHR by setting up an interoperability scenario between the EMR system *i.s.h.med* of Cerner, part of SAP for Healthcare, and the PHR system *HealthVault*, developed by Microsoft. Subsequently we present an integration design for setting up an integrated PHR. The research is concluded with an architecture that implements the integration design into the current Dutch health information exchange infrastructure *AORTA*.

KEYWORDS: Personal Health Record (PHR), Electronic Medical Record (EMR), HL7, SAP for Healthcare, SAP Patient Administration and Billing (PAB), i.s.h.med, Microsoft HealthVault, AORTA, Integration Architecture.

Management summary

This thesis is the result of the research project on the topic of integrated Personal Health Record (PHR) systems. In an exploratory research, we have studied the possibilities towards the integration of PHR systems and Electronic Medical Record (EMR) systems. These systems are defined as:

EMR system "This environment [(an EMR system)] supports the patient's electronic medical record across inpatient and outpatient environments, and is used by healthcare practitioners to document, monitor, and manage healthcare delivery within a Care Delivery Organization (CDO). The data in the EMR is the legal record of what happened to the patient during their encounter at the CDO." (Garets & Davis, 2006)

PHR system "A Personal Health Record is a collection of an individual's lifelong health-related information, that originates from various information sources and is stored in an electronic form conforming to interoperability standards and can be managed, shared and controlled by the individual." (The National Alliance for Health Information Technology, 2008)

This research is executed in collaboration with SAP Nederland B.V. (SAP NL). SAP SE (of which SAP NL is the Dutch subsidiary) develops various software products for the healthcare domain under the name SAP for Healthcare (SfH). One of these products is SAP Patient Administration and Billing (PAB), which is software for CDOs for the administration of all non-clinical activities. The EMR system, which is part of SfH is the clinical extension for SAP PAB, called *i.s.h.med*.

The central research question in this project is '*How can we link an Electronic Medical Record system with an existing free-standing Personal Health Record system in order to set up an integrated PHR?*'. We answer this question by delivering an architecture that supports this integration based on theoretical and practical fundamentals. After an exploratory literature review, we analyzed existing PHR systems in order to have findings on the current state of affairs regarding these systems. Furthermore we investigated the healthcare interoperability standards and how these could contribute to integrated PHRs. In order to gain domain knowledge from practice we did an in-depth analysis on the Dutch health information exchange infrastructure. Furthermore we have built a prototype of an integrated PHR by setting up an interoperability scenario between *i.s.h.med* and the PHR system *Microsoft HealthVault*. This prototype provides us hands-on experience on what challenges arise from this practical perspective. The next

step is to converge our findings to a generic design for realizing an integrated PHR, which is subsequently applied into an architecture, based on the Dutch infrastructure.

Analysis

We have analyzed 68 PHR systems in order to get a clear understanding of the currently existing systems and how these are positioned by their developers. Most of these have a free-standing architecture, indicating that these are fully independent from other systems. Only one PHR could be classified as an *integrated* PHR system, which illustrates that currently integrated PHR systems are uncommon. Two important characteristics for this research regarding PHR systems are the presence of an Application Programming Interface (API), such that interfaces exist to exchange health information, and an extensive support for clinical data types as we have seen that a part of the PHR systems only support a small set of clinical data, e.g. allergies and medication.

The analysis of interoperability standards indicates that the organization Health Level Seven International (HL7) is currently the leading party for the development of interoperability standards for the healthcare domain. HL7 currently develops three interoperability standards:

HL7v2 HL7v2 is initiated in 1987 and is the most widely used standard worldwide. HL7v2 provides standard for various messages and can be described as a flexible, easy to implement message standard.

HL7v3 HL7v3 is initiated in 1997 and is known for the underlying RIM. Every message standard within HL7v3 is derived from the RIM, which ensures a level of consistency. However, HL7v3 is not successful due to its inflexibility, complexity and the abstractness in which it is delivered.

FHIR FHIR is initiated in 2011 and is currently released as DSTU. The key entities of FHIR, called *resources*, are derived from RIM, but in this case the message standard attempts to be less constrained and complex by letting go the ability to cover any piece of health information, but allowing for a proper way to extend FHIR messages. Furthermore FHIR includes a specification for a RESTful API, such that not only the message but also the communication interfaces can be standardized.

The third analysis topic is the Dutch health information exchange infrastructure, called AORTA. In this infrastructure a central system, called Landelijk Schakelpunt (LSP), fulfills the role of a *transport facility*, as it manages the exchange of health information between various systems that are part of the AORTA infrastructure. LSP also acts as the system where a individual can register his consents regarding the exchange of his health records and can access an audit log to check which exchange has occurred in the past. All information exchange, which is currently limited to a medication overview and a summarized health record for general practitioners, in AORTA is based on HL7v3 messages.

Prototype

In the prototype we have implemented the exchange of *weight* measurements between i.s.h.med and HealthVault. HealthVault has proven to be a mature, well-developed PHR system. Due to its platform approach, HealthVault provides an extensive API and allows third parties to develop applications on top of the HealthVault platform. We used the default HealthVault authorization-based access option *Patient Connect* for accessing PHRs and used HL7v2 for a message standard.

A video demonstration of the prototype is available at http://graduation.bastianen.com/. The recommendations that result from the prototype implementation are:

- Ensure consent awareness at the source system.

  This implies that any system that contains health information should be able to determine which data has to be exchanged.

- Use an existing interoperability standard.

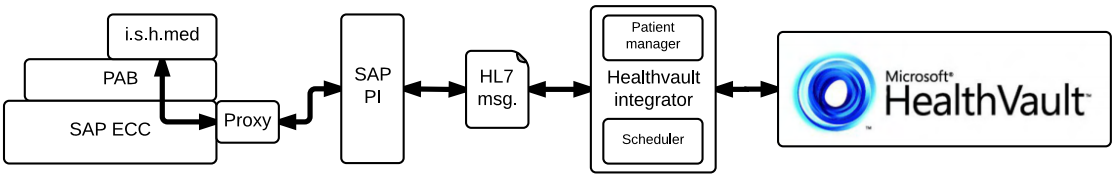
  The use of a standard is inevitable when exchanging information with external parties. When choosing an existing standard, the scalability of the integration scenario is expected to be bigger as potentially more parties are able to comply with the standard.

- Establish event-driven messaging.

  In order to ensure data integrity and to reduce overhead, source systems preferably present new or updated information actively to a target system, i.e. an event-driven approach.

- Secure scenario integrity.

  Due to the complexity, the sensitivity and required integrity in health information exchange, an integration scenario can only have a minimal, manageable risk for errors by design.



Prototype architecture

Integration design

Based on the findings and subsequent design choices, the following design principles have to be considered when realizing an integrated PHR:

Principle 1. *The individual can define which participant is allowed to access his PHR (a consent).*

Principle 2. *The individual can refine a consent such that the participant can access only a part of his PHR.*

Principle 3. *The individual has the ability to revoke his consent.*

Principle 4. *The individual can review audit logs, such that he can verify whether no unauthorized access has been provided to his PHR.*

Principle 5. *The individual can define which parts of his PHR can be accessed when accessing the PHR with an emergency access key.*

Principle 6. *Any actor is uniquely identifiable in order to ensure the authenticity of this actor and to be able to perform proper auditing.*

Principle 7. *Any actor uses an authentication method that is considered to be sufficiently reliable and strong.*

Principle 8. *The integration of data is achieved by setting up a virtual link to the EMRs and PHRs.*

Principle 9. *The individual may still choose to store data from an EMR in his PHR.*

Principle 10. *Any piece of information that originates from a health record is uniquely identifiable.*

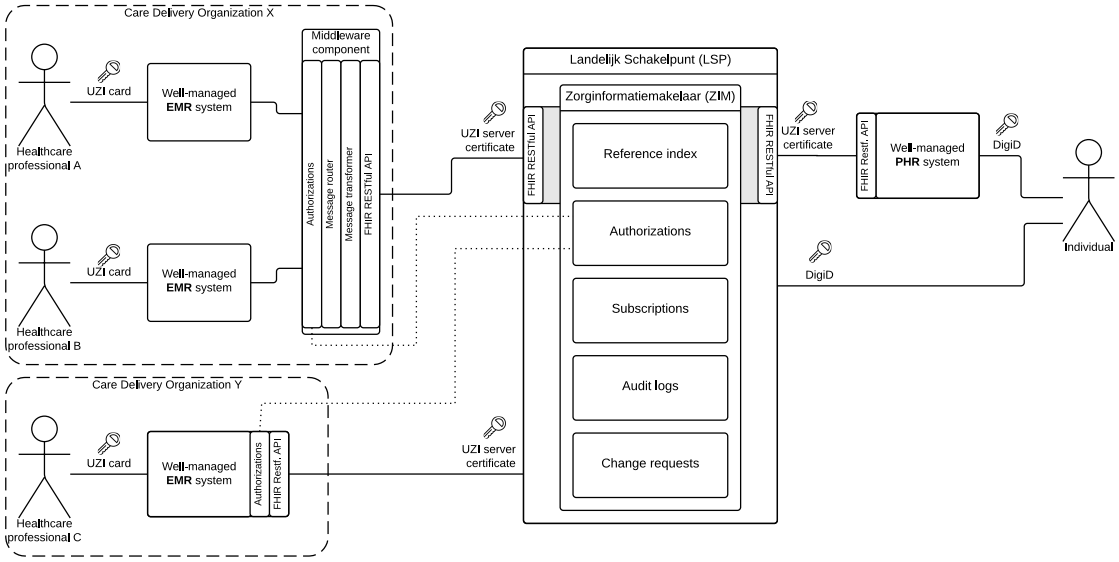
Principle 11. *A neutral party should be in place that decides on which standards are used and how these are optionally extended.*

Principle 12. *Any message that is exchanged complies with a predefined interoperability standard, set by the neutral party.*

Principle 13. *The interoperability standard should not be limited in the sense that participants could be excluded due to the too big effort to support the standard.*

Principle 14. *Any data object can only be modified by its creator. Other participants need to request a change.*

Based on these principles, the Dutch exchange infrastructure AORTA has been extended for the support of integrated PHR systems. By relying on virtual links based on HL7 FHIR, we propose an architecture in which every system provides an implementation of the FHIR RESTful API. LSP still fulfills the central role as transport facility, but is further extended in order to support FHIR. Furthermore LSP contains a more elaborate *authorization* module, such that individuals can provide consents in a more detailed and advanced way. Concluding a new module *change requests* is added to LSP, such that it becomes possible for a participant to modify data that originates from another participant.



Integration architecture overview

Conclusions

The main conclusion that is drawn from this research is that the integration of PHR systems with EMR systems is a non-trivial, but realistic scenario. However, it requires the full cooperation of every involved party regarding decision-making, patient-centeredness, technical support etcetera. With a virtual integration, consent-based approach, concerns regarding patient privacy, data integrity and health record ownership are properly addressed and ensures the success of integrated PHRs. Further research regarding the following topics is required in order to ensure the successful realization of integrated PHRs in the near future:

HL7 FHIR	HL7 FHIR is currently released as DSTU and little experience with this standard is currently available.
A central consent system	The integration design requires an elaborate, central consent system. This system will have to be designed in a preferably standardized manner such that it can be integrated with for example FHIR.
Requirements engineering	Based on the deliverables of this research, a next step would be to start with requirements engineering.
Other aspects	Other aspects, which are not in the scope of this research, require consideration. For example privacy legislation or the financial aspects should be evaluated and incorporated.

Preface

Eindhoven, March 2015

This master thesis marks the end of the graduation project that I carried out at SAP NL from September 2014 to March 2015 in completion of the Master Business Information Systems at Eindhoven University of Technology. This also indicates the conclusion of my time as a student and I would like to take this opportunity to express my gratitude towards those who have helped me during the past years.

First, I would like to thank all colleagues at SAP for helping me during my graduation project. It was a true pleasure to meet the real experts and to have to opportunity to learn from them. In particular I would like to thank Ton Bertram for giving me the opportunity to do my graduation project at SAP. Also a big thanks to Frans Ploegmakers. Frans, I truly enjoyed having you as my daily supervisor and I appreciate that you always took the time to answer my questions and to explain the (sometimes remarkable) world of healthcare IT. In special, I am really grateful for involving me in the healthcare team of SAP and giving me the opportunity to participate in activities that were not directly related to my project.

I would also like to thank my university supervisors, Pieter Van Gorp and Mykola Pechenizkiy, for the support throughout this project. Pieter, thank you for your continuous enthusiasm, elaborate feedback and of course your near real-time e-mail responses. It was a true pleasure having you as my first supervisor. Mykola, I appreciate that you were willing to join as a second supervisor and to took the time to provide me with intermediate feedback. I am sure that it has improved the thesis that I am delivering now.

Then, I would like to say a big thank you to my friends: Those I know from high school and those I have met during my time in Eindhoven. I share some great memories with you and I hope to keep seeing you all in the future.

Last, I would like to express my thankfulness to my parents and brothers for the unconditional support throughout my life. I truly believe that you enabled me to seize any opportunity that I came across, which made me the person that I am today. I conclude with a loving thank you to my girlfriend Marle. Thank you for always being there when I need a hug or a talk. I feel honored having you by my side.

Sven Bastianen

Contents

Abstract iii

Management summary v

Preface xi

1 Introduction 1
- 1.1 Company introduction . . . 1
- 1.2 Problem statement . . . 2
- 1.3 Methodology . . . 3
- 1.4 Research questions . . . 3
- 1.5 Outline . . . 5

2 Literature review 7
- 2.1 Patient-centered healthcare . . . 7
- 2.2 Hospital Information System . . . 7
- 2.3 Electronic Medical Record . . . 8
- 2.4 Electronic Health Record . . . 8
- 2.5 Personal Health Record . . . 9
- 2.6 Wearable Health Monitoring System . . . 10
- 2.7 Interoperability . . . 11
- 2.8 Research contribution . . . 12

3 Analysis 13
- 3.1 PHR systems . . . 13
- 3.2 Standards . . . 18
- 3.3 AORTA . . . 23
- 3.4 Conclusion . . . 26

4 Prototype implementation 29
- 4.1 SAP for Healthcare . . . 30
- 4.2 Microsoft HealthVault . . . 33
- 4.3 Specification . . . 36
- 4.4 Reflection . . . 39
- 4.5 Recommendations . . . 41

5 Integration design **43**
5.1 Actors . . . 43
5.2 Design . . . 43
5.3 Architecture . . . 48
5.4 Roadmap . . . 52

6 Conclusions **53**
6.1 Future research . . . 56

Bibliography **59**

Appendix A Types of PHR system architectures **67**

Appendix B Interoperability patterns **69**

Appendix C PHR analysis results **71**

Appendix D Health Level 7 **77**

Appendix E SAP for Healthcare **89**

Appendix F Microsoft HealthVault **93**

Appendix G Prototype **99**

Appendix H Interview minutes **109**

List of Tables

1.1 SAP SE and SAP NL key numbers 2013 2

2.1 PHR types 10
2.2 Integration patterns (Hohpe & Woolf, 2004) 11

3.1 Primary and secondary drivers for software selection (Sahay & Gupta, 2003) 14
3.2 PHR system characteristics 15
3.3 CDA design principles (Benson, 2010; Health Level Seven International, 2014b) . . . 20
3.4 Comparison of HL7 standards (Bender & Sartipi, 2013) 23
3.5 Actors of LSP (VZVZ, 2013) 24
3.6 Actor authentication methods 25
3.7 Primary interactions of LSP 25
3.8 Supporting interactions of LSP 26

4.1 SAP ECC interoperability options 32
4.2 Authorization-based access options (Microsoft, 2015a) 35

5.1 Actors in the integration design 43

C.1 PHR systems analysis 72

F.1 Microsoft HealthVault API Methods (Microsoft, 2014c) 93
F.2 Microsoft HealthVault Data Types (Microsoft, 2014a) 95

G.1 Sequence diagrams: Systems and components 102

H.1 Interview Robert Duivenvoorden (October 14, 2014) 109
H.2 Interview Ronald de Zoete (November 25, 2014) 111
H.3 E-mail conversation Alf Zwilling (January 30, 2015) 112

List of Figures

1.1 Research process overview 3

2.1 Idealized concept of a PHR (Kaelber et al., 2008) 10

3.1 HL7v3 Body Weight Information Model (Health Level Seven International, 2013) . . 20
3.2 LSP overview 26

4.1 SAP ECC interoperability options 31
4.2 Prototype architecture 37

5.1 Integration architecture overview 50
5.2 Process model for a CDO requesting patient data via LSP 51

A.1 Freestanding PHR system 67
A.2 Provider-tethered PHR system 67
A.3 Claims-tethered PHR system 68
A.4 Integrated PHR system 68

B.1 Interoperability patterns 69

D.1 Common HL7v2 message segments (Benson, 2010) 77
D.2 HL7 Reference Information Model v2.41 (Health Level Seven International, 2012) . . 78

E.1 Product overview of SAP ECC 89
E.2 Product overview of SAP for Healthcare (SAP SE, 2014b) 90
E.3 Modules overview of SAP PAB and i.s.h.med (Siemens AG, 2014) 91

G.1 Patient registration process 99
G.2 Sequence diagram: Register patient 103
G.3 Sequence diagram: Create HL7 message from i.s.h.med observation 104
G.4 Sequence diagram: Send HL7 message to HealthVault 105
G.5 Sequence diagram: Create HL7 message from HealthVault observation 106
G.6 Sequence diagram: Send HL7 message to i.s.h.med 107

List of Abbreviations

A-D-T	Admission-Discharge-Transfer.
ABAP	Advanced Business Application Programming.
API	Application Programming Interface.
BAdI	Business Add In.
BIG	Beroepen in de Individuele Gezondheidszorg.
BSN	Burgerservicenummer.
CCD	Continuity of Care Document.
CCR	Continuity of Care Record.
CDA	Clinical Document Architecture.
CDO	Care Delivery Organization.
CIBG	Centraal Informatiepunt voor Beroepen in de Gezondheidszorg.
CRUD	Create, Read, Update and Delete.
DLL	Dynamic-Link Library.
DSTU	Draft Standard for Trial Use.
ECC	ERP Central Component.
EHR	Electronic Health Record.
EMR	Electronic Medical Record.
ERP	Enterprise Resource Planning.
FHIR	Fast Healthcare Interoperability Resources.
GUI	Graphical User Interface.
HIS	Hospital Information System.
HITECH	Health Information Technology for Economic and Clinical Health.

HL7	Health Level Seven International.
HL7v2	HL7 Version 2.
HL7v3	HL7 Version 3.
HTML	HyperText Markup Language.
HTTP	Hypertext Transfer Protocol.
HTTPS	Hypertext Transfer Protocol Secure.
ICD	International Classification of Diseases.
ICF	Internet Communication Framework.
IDoc	Intermediate Document.
IS–H	Industry Solution Healthcare.
JSON	JavaScript Object Notation.
LAN	Local Area Network.
LOINC	Logical Observation Identifiers Names and Codes.
LSP	Landelijk Schakelpunt.
MDA	Model Driven Architecture.
MLLP	Minimum Lower Layer Protocol.
NFS	Network File System.
NICTIZ	Nationaal ICT Instituut in de Zorg.
PAB	Patient Administration and Billing.
PHR	Personal Health Record.
PKI	Public Key Infrastructure.
PKIO	Public Key Infrastructure Overheid.
PMD	Parameterized Medical Documentation.
PPE	Pre-Production Environment.
REST	Representational State Transfer.
RFC	Remote Function Call.
RIM	Reference Information Model.
RVZ	Raad voor de Volksgezondheid en Zorg.
SAP BC	SAP Business Connector.
SAP JCo	SAP Java Connector.
SAP NCo	SAP Connector for Microsoft .NET.
SAP NL	SAP Nederland B.V..
SAP NRL	SAP Netweaver RFC Library.
SAP PI	SAP Process Integration.
SAP SLD	SAP System Landscape Directory.
SDK	Software Development Kit.

SfH	SAP for Healthcare.
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms.
SOAP	Simple Object Access Protocol.
SODA	Software on Device Authentication.
UK	United Kingdom.
USA	United States of America.
UZI	Unieke Zorgverleners Identificatie.
VZVZ	Vereniging van Zorgaanbieders voor Zorgcommunicatie.
WHMS	Wearable Health Monitoring System.
WHO	World Health Organization.
XML	Extensible Markup Language.
XSD	XML Schema Definition.
ZIM	Zorginformatiemakelaar.

Chapter 1

Introduction

Nowadays the use of Electronic Medical Records (EMRs) for documenting healthcare delivery is widely adopted. EMR systems are typically large enterprise information systems used by hospitals and other Care Delivery Organizations (CDOs) to digitally store clinical data. Next to these, Personal Health Records (PHRs) are an upcoming fashion for individuals to store data about their well-being. Although these systems hold information that is strongly related and can provide each other data that potentially improves the reliability and completeness of the records significantly, little knowledge is currently available on a way to integrate these systems.
Various reasons and arguments can be found that justify the need for information exchange between patients and CDOs (and therefore between a PHR and an EMR). For example the increasing demand of patients to get a more central and informed role in their care delivery process pleads for information exchange with their CDO. Another example is the fact that patients are nowadays creating health-related data through wearables and other smart technology that may be relevant for their physicians. This causes a demand from the CDO for this information exchange.
In this research we aim to deliver an architecture for the scenario of integrated PHRs. This is done by identifying the challenges that such an integration scenario is facing regarding the current state of affairs and the technological effort that is required for integrating multiple systems. Next to challenges found by a literature review, a prototype is developed such that challenges and design choices that arise from a practical perspective are included as well. The delivered architecture is generic in the sense that it is independent from the actual systems that are involved.

1.1 Company introduction

This research is conducted in collaboration with SAP Nederland B.V. (SAP NL). SAP NL is the Dutch regional office of the German corporation SAP SE, one of the largest enterprise software developers worldwide. SAP SE develops software as on-premise or cloud solutions for a wide range of industries and functional areas, e.g. finance, manufacturing or sales, and is market leader in the market of Enterprise Resource Planning (ERP) software. In Table 1.1 some key numbers of SAP SE and SAP NL are stated as an indication of the company's size and performance. One of the solutions that SAP SE offers is *SAP for Healthcare (SfH)*. SfH is a collection of stand-alone or integrated products that are generic and can be used within any organization, i.e. ERP functionality, and products that are dedicated to the healthcare domain. It should be noted that products that are based on SAP Technology do not necessarily have to be owned by SAP SE, but can for example be made by a third party. In Figure E.2 in Appendix E the collection of

Table 1.1: SAP SE and SAP NL key numbers 2013

	SAP SE	**SAP NL**
Revenue	€ 16.8 bln	€ 462 mln
Profit after tax	€ 3.3 bln	€ 31 mln
Customers	263.000	*unknown*
Employees	68.800	447

products that are part of the solution SfH are shown. The core healthcare component is Patient Administration and Billing (PAB), which is owned and developed by SAP NL. PAB fulfills the role of a Hospital Information System (HIS) and provides non-clinical functionality to a CDO. PAB is customized by SAP NL for the Dutch healthcare industry in order to comply with local laws, legislation and guidelines. PAB is used by 12 out of 92 Dutch hospitals. Typically PAB is used in combination with the component *Clinical Treatment and Care* that fulfills the role of EMR system. This component has the name *i.s.h.med* and is not owned by SAP SE, but by the corporation Cerner[1]. i.s.h.med is used by 10 of the 12 Dutch hospitals that are also equipped with SAP PAB (Zorgvisie, 2014).

1.2 Problem statement

The constrained, limited support for standardized interoperability scenarios in the field of healthcare IT bring concerns about how these systems are going to be aligned with the increasing demand for openness of these systems. The importance of this interoperability is growing as the scatteredness of health data is growing and patients are getting more demanding for having access to their health records. Although multiple initiatives have been announced that describe the required openness of health information systems, e.g. SAP's *HANA platform for Healthcare* (SAP SE, 2013) or Epic's *App Exchange* platform (Wisconsin State Journal, 2015), and extensive literature is available on the topic of interoperability, the detailing towards healthcare and the challenges that arise in healthcare interoperability scenarios is not trivial. Especially the patient-centered approach is a relatively new topic in terms of integration design and setup. It remains unclear in what way these developments should be incorporated by EMR and PHR system developers and what the ultimate integration scenario should look like. As SAP NL is investigating the best way to anticipate on this subject, we can argue the central problem and justify this research.

[1] i.s.h.med was owned by Siemens AG, but got acquired by Cerner in August 2014.

1.3 Methodology

The methodology used in this research is derived from the regulative cycle of van Strien (1997). This framework provides a structured approach for solving a business problem. This research starts with a *problem* that we are facing, as stated in section 1.2, and defining the context of this research by composing research questions, objectives and a scope. Subsequently van Strien (1997) prescribes a phase of *analysis and diagnosis*. This phase has resulted in a literature review, additional analyses and the development of a prototype. In this literature review the key elements of this research are explored and defined, e.g. the definition of a PHR. Subsequently an extensive analysis is done in order to gain knowledge on non-scientific side of the problem, i.e. existing PHR systems, existing interoperability standards and the Dutch healthcare information exchange infrastructure. This knowledge will be expanded with the findings that are derived from the learned lessons during the development of a prototype. After validating this knowledge with field experts, an architecture design will be made together with a road map towards this architecture as a final deliverable of this research. van Strien (1997) describes this road map as a *plan of action*. As illustrated, the process is highly iterative. This is mainly caused by the fact that the prototype is mostly build in in parallel to the generic architecture design, which constantly leads to the need of adapting this research as new lessons are learned and may require further analysis. Concluding, we document all results and findings in this thesis. Figure 1.1 points out the steps that have been taken in accordance to what is described in this section.

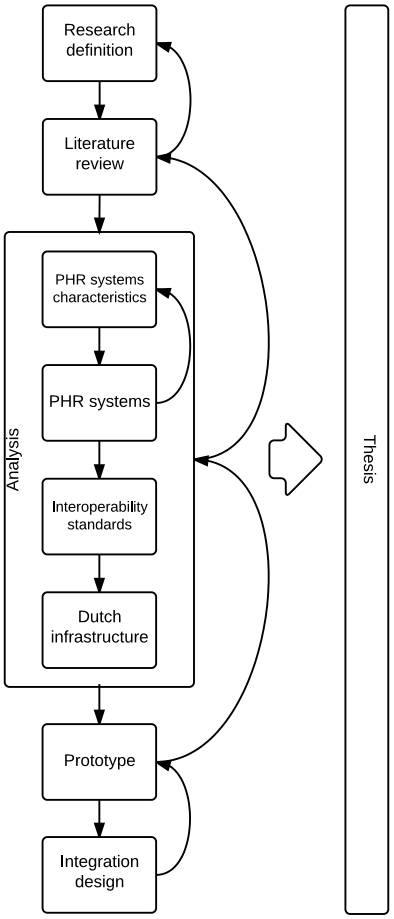


Figure 1.1: Research process overview

1.4 Research questions

This research is structured by the definition of a central research question:

How can we link an Electronic Medical Record system
with an existing free-standing Personal Health Record system
in order to set up an integrated PHR?

The term *system* should be interpreted in the sense of a software information system that comprises the typical three tiers *presentation*, *application* and *data*, as this research focuses on the technical impact of an integrated PHR. The central question is supported by six sub questions that structure the research in more detail. These sub questions have been constantly reconsidered during this research and have been iteratively redefined when necessary.

1: Which PHR systems are currently available and what are their characteristics?

2: How is health-related data currently shared among different parties?

3: What are the lessons learned when prototyping this integrated PHR?

4: Which challenges arise when we integrate a PHR with an EMR?

5: What is the preferred IT architecture in order to realize an integrated PHR?

6: Which steps should be taken in order to realize this architecture?

Scope

In order to increase the relevance and feasibility of this research, this project is subject to a predefined set of constraints and assumptions. These form the scope of this research.

- This research is focused on the architectural and technical areas of the research topic. This implies that this research has a limited focus on other areas, e.g. laws and legislation.
- In case decisions should be made on topics that are country-dependent, the research will be focused on The Netherlands.
- This research is conducted at SAP NL, which makes SAP NL a direct stakeholder of this research. Therefore the HIS *PAB* and EMR system *i.s.h.med* are considered in the prototype.
- In this research multiple PHR systems will be considered and compared, but the prototype will be limited to only the single PHR that gets selected.
- The prototype is considered to be valid when a delimited, logical scenario is covered.

Aim and objectives

The aim of this research is to contribute to the realization of an integrated PHR by providing an architecture for this integration that is based on theoretical and practical fundamentals. In order to achieve this, the following objectives are realized:

- An analysis of existing PHR systems.

  This analysis provides insights about how PHR systems are positioned and what features they have. This analysis supports the selection of a PHR system for the prototype implementation.

- An overview of the healthcare information exchange infrastructure in The Netherlands.

  When we have knowledge on the current Dutch infrastructure that is in place, we are able to reuse (parts of) this infrastructure in the integration architecture that is delivered.

- A prototype implementation of a PHR integrated with the EMR.

  A prototype implementation is used to identify challenges and design choices that arise from a practical sense, in addition of the challenges found through literature.

- A collection of challenges and design choices in the PHR integration design.

  The collection of challenges and design choices gives a clear overview of this aspects have to be covered in the integration design.

- An architecture for the realization of an integrated PHR.

  As a result of the integration design, an architecture explains how the integration scenario looks like in terms of involved systems and how these are connected.

- A road map for the realization of an integrated PHR.

  The road map ultimately indicates what steps have to be taken in order to realize an integrated PHR.

1.5 Outline

This remainder of this thesis is structured as follows: A literature review (chapter 2) is used to provide a formal introduction to the topic of this research and concepts that are related to the topic. Furthermore we position the topic in the practical context by elaborating on the current state of affairs regarding PHR systems and the infrastructure that is already in use for the exchange of health-related data (chapter 3) between different CDOs. In chapter 4 we present the prototype that has been made and elaborate on the design choices and challenges that have been found during the development of the prototype. After this we present the main deliverable in chapter 5, where generic design principles are presented that are subsequently . This thesis is concluded by answering the research questions (chapter 6) and providing directives for future research (section 6.1).

Chapter 2

Literature review

This chapter provides an overview of the academic literature that has been studied to create an initial base for this research. Relevant topics and subjects that are of concern are explained and definitions are given if necessary.

2.1 Patient-centered healthcare

In recent years the healthcare sector is experiencing a shift towards patient-centered care. Although there are various definitions of patient-centered care (Puustjarvi & Puustjarvi, 2010), every definition includes aspects such as a patient's needs and preferences, the need to share information, involvement in decisions of professionals and the need for a wide understanding of the patient's world (Gillespie et al., 2004) in order to design new care models. As described by Daglish & Archer (2009), this patient-centered approach is becoming popular in Western healthcare systems because patients that start to manage their own healthcare may potentially bring lower costs and better outcomes. Little et al. (2001) have even found quantitative, significant evidence for the fact that not having a patient-centered approach can lead to negative effects for the patients' well-being: "If doctors don't provide a positive, patient centered approach patients will be less satisfied, less enabled, and may have greater symptom burden and use more health service resources."

2.2 Hospital Information System

A Hospital Information System (HIS) forms the foundation of a hospital's IT landscape. It provides functionalities to support the administrative processes of a hospital and can be compared to the general concept of an ERP system (Klaus et al., 2000). According to Reichertz (2006) core functionalities of a HIS are:

- Admission-Discharge-Transfer (A-D-T) system, i.e. patient admission.
- Communication system, i.e. the user interface.
- Patient Oriented Administration, i.e. patient billing.
- Finance and Bookkeeping, i.e. accounts payable.
- Material and Disposition, i.e. pharmacy administration.

- Personnel Information Systems, i.e. payroll.
- Organization, i.e. personnel scheduling.

This outlines that the functions of a HIS do not concern the administration of clinical data, i.e. diagnosis, treatment or medication data. This administration is part of an EMR. Ammenwerth & Winter (2004) provided a clear, generic definition of a HIS:

> **Definition 1.** *"A HIS is the socio-technical subsystem of a hospital, which comprises all information processing as well as the associated human or technical actors in their respective processing roles. Typical components of hospital information systems are enterprise functions, business processes, application components and physical data processing components."*

2.3 Electronic Medical Record

One of the biggest changes in the past decades in supporting services at CDOs is the shift from paper-based patient files to electronic records of patients. Care providers started to implement Electronic Medical Record (EMR) systems, which enable them to process patient records in a digital fashion.
An EMR is not defined unambiguously in literature (Koeken, 2012). Liu et al. (2013) describe an EMR as a collection of past, present and future physical and psychological status records of patients. Sonoda (2011) indicates that most EMR systems that are deployed handle patient information within a single medical institution. This is in accordance with the definition of Garets & Davis (2006), which is used in this research:

> **Definition 2.** *"This environment [(an EMR system)] supports the patient's electronic medical record across inpatient and outpatient environments, and is used by healthcare practitioners to document, monitor, and manage healthcare delivery within a CDO. The data in the EMR is the legal record of what happened to the patient during their encounter at the CDO."*

An EMR system gives great opportunities regarding improving efficiency and quality within healthcare, which is also recognized by federal parties. In 2009, the US government enacted the Health Information Technology for Economic and Clinical Health (HITECH) act, which stimulates the use of EMR systems in healthcare institutions by offering financial incentives when implementing such a system (Buntin et al., 2011).

2.4 Electronic Health Record

Next to an EMR, an Electronic Health Record (EHR) exists as well. Although there is no general consensus about the actual distinction between these terms in literature (Hinman & Ross, 2010; Sonoda, 2011; Koeken, 2012), in this research a clear distinction is made. An Electronic Health Record (EHR) can be seen as an extended version of an EMR that spans across multiple CDOs and covers multiple "episodes of care" (Garets & Davis, 2006). Häyrinen et al. (2008) define an EHR as "a repository of patient data in digital form, stored and exchanged securely, and accessible by multiple authorized users. It contains retrospective, concurrent, and prospective

information and its primary purpose is to support continuing, efficient and quality integrated healthcare." This implies that a patient does not have access to his EHR by default. This definition supports the definition of The National Alliance for Health Information Technology (2008), which is used in this research:

> **Definition 3.** *"[An EHR is] an electronic record of health-related information on an individual that conforms to nationally recognized interoperability standards and that can be created, managed, and consulted by authorized clinicians and staff across more than one healthcare organization."*

The Dutch government has attempted to introduce a nationwide EHR system, which resulted in lots of resistance from the Senate of the Dutch parliament, doctors and citizens (Pluut, 2010). After a period of discussion and redesigns, a private association (Vereniging van Zorgaanbieders voor Zorgcommunicatie (VZVZ)) has adopted and further developed the IT infrastructure, called AORTA, that initially had been developed by the Dutch government. This IT infrastructure enable CDOs to share parts of EMRs on the condition that the concerned patient has given his permission in order to have a kind of hybrid EHR.[1]

2.5 Personal Health Record

Combining the shift towards patient-centered healthcare and the possibilities that an EHR system offers, brings that patients start to demand more control over their medical records. In 2008 approximately seventy million US citizens had access to a so-called Personal Health Record (PHR) (Kaelber et al., 2008). A PHR system is described by Kaelber et al. (2008) as "a set of computer-based tools that allow people to access and coordinate their lifelong health information and make appropriate parts of it available to those who need it." PHR is described by Liu et al. (2013) as: "A PHR is an individual's electronic record of health-related information that conforms to nationally recognized interoperability standards and that can be drawn from multiple sources while being managed, shared and controlled by the individual." In this sense, a PHR is a subset of an EHR, which can be controlled by the individual.
In this research a hybrid version of these two definitions is used:

> **Definition 4.** *"A Personal Health Record is a collection of an individual's lifelong health-related information, that originates from various information sources and is stored in an electronic form conforming to interoperability standards and can be managed, shared and controlled by the individual."*

A PHR system offers a patient the possibility to access his health record. A health record can hold different types of information, varying from medication prescriptions to laboratory results or doctor appointments. This data is created and maintained by healthcare providers (by extracting data from the patient's EMRs), insurance companies or the patient himself (L. Chan, 2009). Kaelber et al. (2008) describes four different architectural types of PHRs, which are stated in Table 2.1. A simplified, schematic model of every architecture is illustrated in Appendix A.

[1] More information about AORTA is provided in section 3.3.

Table 2.1: PHR types

Free-standing	A free-standing PHR is an independent system in which a patient enters his health data.
Provider-tethered	A provider-tethered PHR is a system, which is an extension of the EMR system of a care provider, in which a patient can log in and view his data. It is often called a *patient portal*.
Claims-tethered	A claims-tethered or payer-tethered PHR is comparable to the provider-tethered type, but is then connected to the system in which the claims are stored, i.e. the system of an insurer.
Integrated	In the integrated type, the PHR can be seen as a central hub and is interconnected to different information systems of different care providers. As this PHR contains information from different sources, it becomes more valuable than any individual source. Figure 2.1 illustrates this architecture. In this architecture it is not necessarily required that the central PHR node physically stores the data of all the connected systems, but can also act like an aggregated portal that maintains virtual links with the connected systems. At all time, a PHR will provide real-time access to the relevant data.

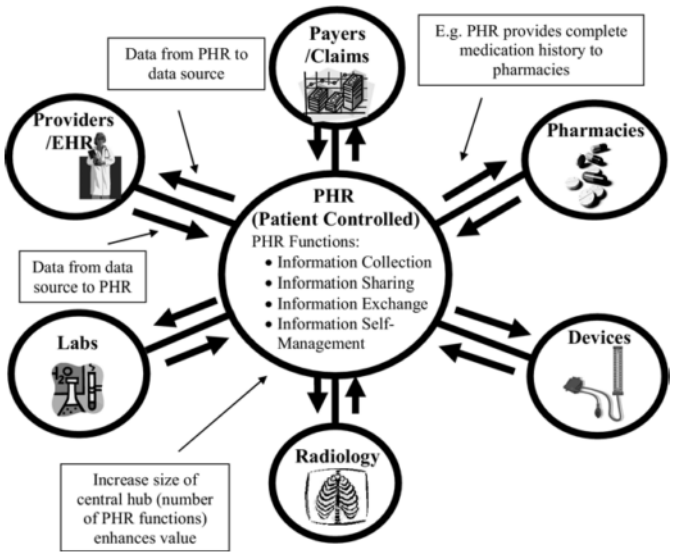


Figure 2.1: Idealized concept of a PHR (Kaelber et al., 2008)

2.6 Wearable Health Monitoring System

In the past years, new technological developments have increased the possible applications of Wearable Health Monitoring Systems (WHMSs). Commercial parties in the high-tech industry are releasing more products that are able to measure certain physiological conditions, such as intelligent blood pressure meters, heart rate monitors or weighing scales. A WHMS opens up new possibilities regarding health monitoring. For example heart patients that are visiting their physician on a frequent base for a routine check. With new technologies all measures and tests

that a physician performs during a routine check can potentially be conducted by the patient himself using a wearable health monitoring system (Khan, 2013).
However, most vendors of WHMSs tend to use the data that is generated by their WHMSs for commercial purposes, i.e. Nike with its WHMS 'Fuelband' (Nike, 2014). This causes that most WHMSs are storing its measurements in a separate system, owned by its vendor, which causes that health data is again scattered. In order to increase the usability of this data, some vendors of WHMSs provide support for other services, i.e. integration with a PHR system. However the supported PHR systems differ between the various vendors so it is not trivial that the data that is generated by a WHMS can be used in the sense of an integrated PHR by an individual. This is strengthened by the fact that although some integration possibilities could be in place between the data store of a WHMS and a PHR system, there is still a lack of standardization about how this integration should be realized.

2.7 Interoperability

Interoperability is the ability of systems to collaborate and aims to boost the common understanding of data, information and functional operations among these (Calvillo-Arbizu, 2014). From a high-level architectural perspective four different approaches can be identified for establishing a link between two systems, as described by Hohpe & Woolf (2004). These patterns are described in Table 2.2. A visual representation of these patterns is given in Appendix B.

Table 2.2: Integration patterns (Hohpe & Woolf, 2004)

File Transfer	"Have each application produce files of shared data for others to consume and consume files that others have produced."
Shared Database	"Have the application store the data they wish to share in a common database."
Remote Procedure Invocation	"Have each application expose some of its procedures so that they can be invoked remotely, and have each application invoke those to initiate behavior and exchange data."
Messaging	"Have each application connect to a common messaging system, and exchange data and invoke behavior using messages."

There are numerous factors that may lead to the selection of a certain pattern. Depending on the characteristics of each system and which techniques are supported, the requirements for the integration, e.g. synchronous integration, and the required effort for realizing the integration, a pattern will be selected. Hohpe & Woolf (2004) are giving a proper explanation of the advantages and disadvantages of these patterns, indicating that integration is all about making separate systems as timely as possible while ensuring that the systems are not coupled too extensively. In case of using a *File Transfer* pattern, systems remain very uncoupled, which as an advantage as interdependence should be avoided. However all timeliness is lost in this scenario, which results in (temporary) out-of-sync systems. Using a *Shared Database* the systems are relying on the same data set, which makes the systems as timely as possible, but also fully ties these systems together on the database level. A big disadvantage of the file transfer pattern and the shared database pattern is that these systems cannot reuse each others functionality, but only

collaborate in terms of data. Sharing functionality can be realized with the *Remote Procedure Invocation* pattern, as systems are then invoking functions or procedures in a remote system. The downside of a system that directly invokes another system is the fact that the scenario is not scalable and it requires that both the sender and receiver systems are online. Therefore an integration is ideally done in a *messaging* pattern, as it provides "the best of both worlds": Timely, asynchronous communication in a decoupled setting.

2.8 Research contribution

The problem statement, aim and objectives, are described in chapter 1, argue the central topic of this research. In this section we position the topic in literature and further clarify its contribution. As indicated the importance of patient-centered care is elaborated based on the significant results in the paper of Little et al. (2001). This significance is confirmed by the literature review of Dorr et al. (2007). In this review the majority of the articles reported positive results on the use of PHRs, which results in an increase in patient involvement and the related improvement in health outcomes. Furthermore one of the four priority areas of Nationaal ICT Instituut in de Zorg (NICTIZ), the Dutch IT institute for healthcare, is *patient empowerment*, which they define as "creating the possibility for patients to have more control in their own care process" (Nictiz, 2014a). This ambition is aligned to the current trend that patients themselves want to have more control, as described by Archer et al. (2011). By disclosing information, e.g. test results or medication prescriptions, from for example a hospital's EMR to the patient via an existing PHR system, the expectation is that this will improve the quality of healthcare from a patient perspective and will put the patient more in control regarding the healthcare services he receives. Furthermore healthcare quality can be improved when it is possible to provide a hospitals EMR with additional data about a patient's health by communicating data that is stored in a PHR. This could enable physicians to make more elaborate decisions about a patient and the healthcare services he receives.
Another potential benefit of an integrated PHR is the upcoming development of self-monitoring and WHMSs. As current technology is improving and the possibilities of a WHMS and other self-monitoring devices are becoming more advanced, the application area of these devices is more broad than in the past. These devices are able to generate lots of data and some of them are even capable to store this data directly into a PHR. For example Microsoft currently lists 233 devices that support their PHR *HealthVault* (Microsoft, 2014b). Apart from the value to the patient, this data can be valuable for care providers as well. For instance early disease detection or patient monitoring (Mukherjee et al., 2014; M. Chan et al., 2012) can be supported with this data.
Apart from the high-level trends and the relevance of PHR, this research will also contribute in the practical sense that currently no integrated PHR exists. By delivering a prototype integration of a free-standing, existing PHR with a hospital's EMR, which is potentially a more valuable environment than a tethered PHR or having a free-standing PHR (Kaelber et al., 2008; Van Gorp & Comuzzi, 2012), this prototype will provide insights and lessons as such an integration has not been described in literature before. Furthermore this research will contribute in the sense that is provides an architecture that is detailed to the scenario of an integrated PHR accounting for challenges that arise from both a theoretical and a practical perspective, but is independent on the actual involved systems.

Chapter 3

Analysis

In this chapter we analyze the current state of affairs regarding PHR systems and healthcare interoperability. These analyses will support the prototype (chapter 4) implementation and the integration design (chapter 5) by having a clear understanding of the current practicalities regarding these two topics. Starting with setting up analysis/selection criteria for PHR systems, a set of existing PHR systems have been checked according to these. Subsequently we describe the current practice regarding healthcare interoperability standards, e.g. Health Level Seven International (HL7) standards. Concluding the current health information exchange infrastructure in The Netherlands is described.

3.1 PHR systems

In order to have a proper understanding of Personal Health Record (PHR) systems, currently available systems are analyzed in order to know which systems exist, where these differentiate, what functionality is common and how these systems are positioned. This chapter describes the analysis of various PHR systems. This analysis is started with the determination of the characteristics of a PHR regarding its usability in this research. Subsequently we searched the internet for *Personal Health Record* systems. When doing so, a lot of existing PHR systems can be found, varying from small local initiatives to large enterprises that offer a PHR system. These systems have been investigated together with existing analyses, e.g. the analysis carried out by Nederlandse Patiënten Consumenten Federatie (2014). In order to restrict the analysis, the analysis is considered to be complete when more than fifty PHR systems have been analyzed. Eventually this analysis is used to select a PHR system for the prototype implementation, as described in chapter 4.

Characteristics

To ensure that a sufficient number of characteristics of PHR systems is covered to be able to select the most appropriate PHR, the framework of Sahay & Gupta (2003) is used. This framework addresses a method for software selection and distinguishes primary and secondary drivers. Within the primary drivers several categories can be distinguished as stated in Table 3.1. Based on the primary drivers, we defined a set of characteristics, as stated in Table 3.2, that are considered to be relevant for a PHR system comparison and selection. For the driver *customization*, no characteristics have been defined due to the fact that it is hard to capture a classification for the degree of customization that is offered by a PHR system. Furthermore it can be discussed

Table 3.1: Primary and secondary drivers for software selection (Sahay & Gupta, 2003)

Primary drivers	Technology
	Cost and Pricing
	Features
	Customization
	Support & Services
Secondary drivers	Vendor Vision
	Industry Covered
	Vendor Strength
	Others

whether customization is actually desirable as customization is typically complicating standardization. Therefore we do not consider the customization capabilities of a PHR system and only consider the standard set of functionality in the PHR system analysis. Due to the abstract character of the secondary drivers, it is hard to define explicit classification for these. However the secondary drivers provide good points of consideration when you have to choose among various PHR systems that are considered equal when evaluating the primary drivers.
The characteristics definition has been iteratively refined when a relevant new selection characteristic was found while analyzing the PHR sytems. For example the characteristic *CCR/CCD support* has been added when we found that some PHR system support this functionality, which could be relevant for this research.

Table 3.2: PHR system characteristics

Technology

1. Architecture
    a) On-premise
    b) Cloud
2. API
    a) Open
    b) Closed
    c) None
3. PHR type
    a) Free-standing
    b) Provider-tethered
    c) Claims-tethered
    d) Integrated

Cost and Pricing

1. Pricing model
    a) Free
    b) Free with paid upgrades (recurring)
    c) Free with paid upgrades (one-time)
    d) Paid (one-time)
    e) Paid (recurring)
    f) Sponsored

Features

1. Clinical data support
    a) Yes
    b) No
2. CCR/CCD support
    a) Yes
    b) Export only
    c) Import only
    d) No
3. Microsoft HealthVault integration
    a) Yes
    b) No

Support & Services

1. Data owner
    a) Patient
    b) Provider
    c) Vendor
2. Dutch language support
    a) Yes
    b) No
3. Open developers community
    a) Yes
    b) No
4. Status
    a) Active
    b) Discontinued
    c) In development

Characteristics

This section gives an explanation for every characteristic and describes how each characteristic is analyzed. These explanations are stated in the list below. Every PHR system is evaluated based on the specifications stated on the website or by actually using the concerned system.

Technology

1. Architecture

   Although there is no clear definition of a cloud-based system (Vaquero et al., 2008), a cloud-based system is considered to be a system that is accessible via the internet and is hosted at the premises of a third party, while an on-premise system is stored by the patient locally.

2. API

   An Application Programming Interface (API) is a module of some software package that offers a standardized interface for external applications in order to interoperate. An open API is a public interface that can be used by an external application. When the API is closed, the PHR is equipped with API, but its use is only available for a set of application developers.

3. PHR type

   The PHR types as described in Table 2.1.

Cost and Pricing

1. Pricing model

   The pricing model indicates what the involved costs are for a patient to use the system. Some PHR systems are offered for free, as the vendor will have a different business model to earn its money, i.e. by using the data of the patient. *Free with paid upgrades* indicates a system that offers a limited set of functionality for free and the user can buy additional functionality for a recurring or one-time fee. *Paid* systems are either requiring a recurrent fee or only an initial fee. *Sponsored* PHRs indicate that the subscription/license of the user is paid by someone else, i.e. the hospital, the employer or insurer.

Features

1. Clinical data support

   This characteristic indicates whether the concerning PHR is capable of storing clinical data, e.g. vital sign measurements and lab results. As some PHRs are only capable of storing a limited set of information, e.g. blood type and allergies, this is a relevant characteristic.

2. CCR/CCD import-export support

   Continuity of Care Record (CCR) is a worldwide recognized standard for the communication of patient information between various care providers (Nictiz, 2012b). Continuity of Care Document (CCD) is a standard that is based on CCR and incorporates HL7 [1] technologies.

[1] HL7 is an organization that provides a framework and standards for the exchange of electronic health information (Health Level Seven International, 2014a). More information about HL7 can be found in section 3.2.

3. Microsoft HealthVault integration

   Microsoft HealthVault is a PHR system with a platform design. This indicates that Microsoft has a focus on external parties developing applications on top of its platform rather than delivering an application themselves. This enables for example other PHR systems to integrate HealthVault with their system.

Support & Services

1. Data owner

   The owner of the data that is stored in a PHR.

2. Dutch language support

   This characteristic indicates whether the Dutch language is supported in the user interface of the PHR system.

3. Open developers community

   This characteristic indicates whether there is an active community of developers for that particular PHR system that is open for developers to join in order to give and receive support to other developers.

4. Status

   This characteristic indicates whether the PHR system is currently actively supported and released by its vendor. Other options are that the system is still in development and only announced or that the system is already discontinued.

Results

In total 68 PHR systems are examined in this analysis and their characteristics have been collected. All results are stated in Table C.1 (Appendix C). A main result of the analysis of PHR systems is the fact that currently a lot of systems are available. From the selection of free-standing PHR systems that have been analyzed various differences have been found, i.e. different architectures or interoperability possibilities. The main purpose of the systems is to provide a personal health data store to people that want to keep track of this data.
All examined PHR systems have a cloud-based architecture, which means that data is not saved at the patient locally, but stored on a server of the PHR provider. The majority of the free-standing systems offer a way to disclose health data to external parties, i.e. a CDO. However, most systems provide this service by enabling a physician to log in to this PHR system and providing him access to the PHRs of a patient. This access can either be provided based on a predefined relationship between the physician and the patient, e.g. *Patiënt1*, or the access is granted by a so-called emergency key, e.g. *MedicAlert*. Such an emergency key is often a hard-copy code that a patient carries with him. This code can be used to bypass the regular authentication process of the PHR system and to obtain immediate access to the PHR. Clearly these ways of information disclosure are not appropriate for the setup of an integrated PHR, due to the fact that the respective data can only be accessed via a Graphical User Interface (GUI) of the PHR provider. Furthermore there is no standard interface to add data to the PHRs.
Some PHR systems are equipped with an API, e.g. *Patients know best*, which enables external parties to interoperate with the system in a predefined fashion. Such an integration interface is required when we want to evolve a free-standing PHR system to an integrated one as data exchange between different systems is one of the main characteristics of an integrated PHR. The

analyzed systems also differ in terms of clinical data support. Some PHR systems are designed to support the registration of basic health data like medication usage or allergies, e.g. *Zorgdoc*. Others, e.g. *Microsoft HealthVault*, have an extensive set of data types that are supported, i.e. measurements, lab results, procedures. The results are used in chapter 4 for the selection of an PHR system for our prototype.

3.2 Standards

A lot of general and healthcare-specific standards exist, serving different purposes and supporting different processes. The big number of existing standards complicates the making of a complete survey of all standards. In this section we discuss existing standards that are widely adopted and show relevance with respect to the research topic.

ICD

International Classification of Diseases (ICD) is a classification system for diseases, maintained by World Health Organization (WHO). The tenth version of ICD contains approximately 32,000 diagnosis codes (Nictiz, 2012b). The main characteristic of this standard is its top-down tree structure: Starting from a high-level classification, e.g. "XIX Injury, poisoning and certain other consequences of external causes", the diseases can be further specified on a detailed level, e.g. "S64.3 Injury of digital nerve of thumb" (World Health Organization, 2015). Starting from July 1st 2015, Dutch CDOs are required to classify every disease that they reimburse with an ICD-10 code in order to standardize this data (Nederlandse Zorgautoriteit, 2014). This increases the usability of this data, i.e. for analysis purposes.

LOINC

Logical Observation Identifiers Names and Codes (LOINC) standardizes the concepts of lab requests, lab results and other clinical terms. By using LOINC, parties are able to exchange data about health-related measurements/observations in a standard manner. A code out of the LOINC terminology consists of six main fields that define the meaning of the code (Nictiz, 2012b): Component, kind of property, time aspect, system, type of scale and type of method.

SNOMED CT

Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) is a terminology standard which aims to cover a practically every concept in healthcare, e.g. symptoms, diagnoses and results, and all related concepts that can support or detail a healthcare concept, e.g. a list of organisms, physical forces. The main characteristic of SNOMED CT is its hierarchical approach, which enables to link various concepts, terms and/or synonyms to each other. This hierarchical approach makes SNOMED CT a very powerful terminology standard as it has the ability to link various terms and concepts such that one can describe an event in much detail. For example, it is theoretically possible to describe the event "A male person got injured due to an electric current caused by industrial electrical equipment, namely a ski lift, while there was a high humidity." based on SNOMED CT terminology. This illustrates the wide range of concepts that is covered by SNOMED CT. However, it is less straightforward to use SNOMED CT for analysis purposes as it is for example hard to aggregate data that is classified with this standard, in contrast to ICD.

HL7

HL7 is a not-for-profit organization that develops the world's most widely used message standards for the healthcare domain. It creates standards for the exchange, management, and integration of electronic healthcare information for clinical and administrative purposes (Benson, 2010). Because the broad collection of messages that is available within the HL7 standards, incorporating other standards, e.g. ICD, LOINC and SNOMED CT for coding purposes, an HL7 message standard is available for almost every information exchange scenario. Currently HL7 has released two standards, namely *HL7 Version 2 (HL7v2)* and *HL7 Version 3 (HL7v3)*. HL7v2 is currently the most widely used standard and build from an ad-hoc approach, which indicates that the standard is based on the data that has to be exchanged from a practical point of view rather than having a theoretical, domain model based approach (Bender & Sartipi, 2013). Because of this approach, HL7v2 does not form a consistent collection of messages. This has lead to the development of HL7v3 where a top-down approach is used by having a central information model from which the message standards are derived. However, due to the lack of adoption of HL7v3, HL7 has started with the development of a third, new standard, called Fast Healthcare Interoperability Resources (FHIR), which is currently published as Draft Standard for Trial Use (DSTU). The standards are described in more detail in the remainder of this section.

HL7 Version 2

HL7v2 was the first released collection of message standards developed by HL7 that are formatted pipe-delimited or in Extensible Markup Language (XML). The success of HL7v2 can be awarded to the fact that this standard still offers a big flexibility. Furthermore documentation and implementation guidelines can be used directly by integration developers and lots of reference implementations are available. HL7v2 covers a lot of different message types, which share common segments, e.g. a message header and a patient identifier. These common segments are stated in the diagram in Figure D.1 in Appendix D. The Version 2 messages contain required and optional segments and fields. The optional segments and fields allow users to put additional data in the message in case the required ones do not offer cover all data that needs to be exchanged. Furthermore it is allowed to add so-called *Z-segments* to a message. Z-segments are custom segments in order to provide a message with additional information that cannot be put in one of the segments provided by HL7 (Benson, 2010). Due to this flexibility, different implementations of HL7v2 messages may exist, which complicates the process for aligning the communication interfaces among different parties (Health Level Seven International, 2014c).
For the transmission of HL7 messages, the protocol Minimum Lower Layer Protocol (MLLP) has been developed. On local networks it is common to transfer messages over the TCP/IP layer of the network. Due to the fact that network traffic over TCP/IP is a continuous stream of data, one may not be able to identify the beginning or ending of a message. MLLP adds a header and trailer characters to each message, such that a receiving system is able to distinct various messages.

HL7 Version 3

In order to overcome the difficulties of HL7v2, HL7 has developed HL7v3 in which a fundamentally different approach is used. Every message standard within HL7v3 is derived from the Reference Information Model (RIM) and is formatted in XML. RIM is a model of the collection of entities and their relations that aims to cover all aspects of a CDO's clinical and administrative information (Eggebraaten et al., 2007). This means that ideally every piece of information and every type of message can be derived from the RIM by taking the affected entities and their

related attributes. By having this model-based, static starting point for every HL7v3 message, the potential level of standardization and interoperability is theoretically higher. An example of the application of this model-based approach is stated in Figure 3.1, which shows the implementation of RIM for a *Body weight* measurement. In Figure D.2 in Appendix D the graphical representation of the current version of RIM is presented.

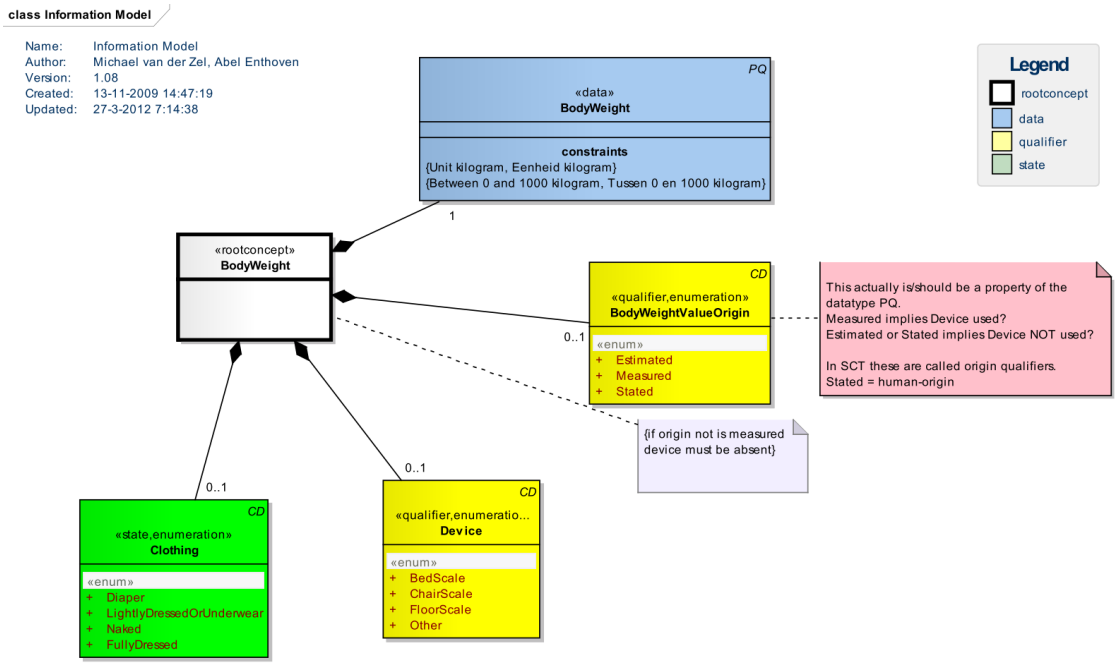


Figure 3.1: HL7v3 Body Weight Information Model (Health Level Seven International, 2013)

Table 3.3: CDA design principles (Benson, 2010; Health Level Seven International, 2014b)

Persistence	"A clinical document continues to exist in an unaltered state, for a time period defined by local and regulatory requirements."
Stewardship	"A clinical document is maintained by a person or organization entrusted with its care."
Potential for authentication	"A clinical document is an assemblage of information that is intended to be legally authenticated."
Wholeness	"Authentication of a clinical document applies to the whole and does not apply to portions of the document without the full context of the document."
Human readable	"A clinical document is human readable."

Another corner stone of HL7v3 is the Clinical Document Architecture (CDA). CDA is an abstract message standard that is based on the principles described in Table 3.3. A requirement of CDA is human readability of a message, such that a message is not just for machine processing purposes. This is achieved by including the health information in plain text, HyperText Markup Language

(HTML) or any browser-interpretable format as HL7 requires that it should be readable by using "a standard Web browser" (Health Level Seven International, 2005). Remarkably, CDA does not even enforce the need for structured, machine-interpretable data in a message, but structured data sections can be optionally added in order to present the unstructured data in a machine-interpretable format. This structuring is done using the RIM and results in a single message that can be used for human and machine processing. An example of a CDA-based message is given in Listing D.1 in chapter D. In this XML-message a human-readable section, i.e. the section `<text>...</text>`, and a machine-interpretable section, i.e. the section `<entry>...</entry>`, is included. Due the abstractness of CDA, checking whether some HL7v3 message conforms to the CDA is currently not fully possible by a machine, but requires human reading and interpretation. A base check can be done by removing any extension from a HL7v3 message and validating it to the base schema of CDA. However this will only ensure that the message is in a proper XML format. For example, it depends on the type of message and the underlying specification what coding system, e.g. ICD or SNOMED CT, is required for some field or segment. Furthermore it is difficult to come up with an automatic validation mechanism that checks whether all information that is provided in the machine-interpretable part of the message is also present in the human-readable part. This complicates the implementation of proper, valid CDA-based communication.

CCR/CCD The best known application of CDA is the CCD message standard, which is based on the earlier developed message standard CCR. CCR is developed by various healthcare institutions for describing a health summary of a person based on a core set of seventeen segments:

- Header
- Purpose
- Problems
- Procedures
- Family history
- Social history
- Payers
- Advance directives
- Alerts
- Medications
- Immunizations
- Medical equipment
- Vital signs
- Functional stats
- Results
- Encounters
- Plan of care

The biggest part of the CCR message structure concerns optional segments, which makes the standard relatively open and flexible. Furthermore CCR does not keep track of relations between various segments, i.e. which diagnosis/problem has lead to which medication (Nictiz, 2012a). The CCD standard is the implementation of CCR into CDA developed by HL7. By mapping the entities of CCR to the RIM, it incorporates the value of CCR and the structure and human interpretability of CDA, which makes the CCD standard a common choice for the communication of a complete health record or a "snapshot" of this record. In Appendix D an example is given of a CCD message (Listing D.2) containing vital signs. Furthermore examples are given of the equivalent HL7v2 ORUˆR01 message in Listing D.3 and Listing D.4. From these examples we may state that in the situation where one wants to exchange one vital sign measurement, the CCD format will cause a lot of overhead in terms of required information in the message. Using the message standard that is dedicated for a vital sign would then be a more appropiate choice.

Fast Healthcare Interoperability Resources

HL7 recognizes that HL7v3 is not widely adopted in the domain of healthcare interoperability. The main reasons for this is the complexity of this standard together with the fact that the standard is too abstract and conceptual and still requires a lot of definition and detailing when some party wants to implement it. In 2011 HL7 started with the development of FHIR, which would be a new standard that embeds the advantages of HL7v3 together with the ease of implementation of HL7v2. The vision of HL7 when developing FHIR is not to deliver a standard in which every possible piece of information can be structured, but to support the common parts of health information that have to be exchanged while providing a structured method to extend a message format to the need of a user. This is an important characteristic that is fundamentally different than the approach of HL7v3.
The fundamental concept in FHIR is a *resource*. For key entities within health information exchange a resource has been created, e.g. patient, device and observation. Most of these resources are derived from the RIM, but not every resource. HL7v3 is designed such that the RIM is an actual constraint, while in FHIR resources are derived from RIM, but the standard is not constraint to the limitations or the scope of RIM. (Health Level Seven International, 2014f). The main difference between HL7v3 and FHIR is the level of abstraction in which the standard is delivered to the parties that implement the actual interoperability scenarios. Due to the fact that FHIR and the related documentation can be directly used for implementation makes this standard easier accessible for developers. In Table 3.4 the main differences between HL7v2, HL7v3 and FHIR can be found[2].
Until FHIR HL7 only addressed the standardization of messages and their formats. A new focus area of HL7, that is part of the success strategy of FHIR, is how messages are transferred. They believe that common web technology should be more incorporated in their standards. The FHIR standard includes a full specification of a RESTful API. Representational State Transfer (REST) is an approach for exchanging data by the pure use of Hypertext Transfer Protocol (HTTP) and its methods *GET*, *POST*, *PUT* and *DELETE*. With FHIR, HL7 attempts to demonstrate that healthcare interoperability could be improved when every involved system can be accessed over the same set of interfaces, i.e. the RESTful API. Furthermore FHIR explicitly allows messages to be formatted in JavaScript Object Notation (JSON) next to XML formatting. JSON is a simple data format that consists of value-attribute pairs and is a common used format in web service based data transmissions. Although HL7 explicitly describes this RESTful API to use in combination with the FHIR messages, this does not exclude other messaging mechanisms such as Simple Object Access Protocol (SOAP).

[2]Further information on the differences between FHIR and other HL7 standards can be found in Health Level Seven International (2014d).

Table 3.4: Comparison of HL7 standards (Bender & Sartipi, 2013)

Property	**HL7v2**	**HL7v3**	**FHIR**
Year initiated	1987	1997	2011
Development Process/methodology	Bottom up / ad hoc	Top-down, MDA	Iterative and incremental
Architectural paradigm	Message, Fields and records	Message-Oriented	RESTful
Semantic Ontology	No	Yes	Yes?
Learning overhead	Order of weeks	Order of months	Order of weeks
Specialized tooling required?	Yes - parser	Yes - model compiler	No
Directly consumable?	yes	no	yes
Order of size of specification	hundreds of pages	thousands of pages	hundreds of pages
Implementation examples in specification	yes	minimal	yes
Reference implementations available from HL7	no	no	yes
Industry and community support	strong	weak	n/a - too new
Inherently suitable for mobile devices	no	no	yes
Number of message types	*unknown*	450	30
Degree of adoption	Very high	Very low	n/a
Information model type	ad hoc	constrained RIM	*unknown*
International character support	no (ASCII) conceptually	yes	yes (UTF8)
International message format support	single global standard	localized by realm	single global standard

3.3 AORTA

The Dutch healthcare information exchange infrastructure is called *AORTA* and is maintained by the organization VZVZ. As indicated in section 2.4, the Dutch government has unsuccessfully tried to introduce a nationwide EHR. The government started in 2005 with the development of an IT infrastructure, called *Landelijk Schakelpunt (LSP)*. After a period with a lot of resistance, the bill, describing this EHR, got officially rejected by the Senate of the Dutch parliament in April 2011 due to the questionable security and privacy protection of this system. However,

several healthcare providers have decided to continue the development of LSP privately as part of *AORTA* (Ministerie van VWS, 2012). Although this continuation faced again a lot of resistance, it was decided by the Dutch court that no laws were violated by this system in July 2014 (Rechtbank Midden-Nederland, 2014).
LSP forms the core platform of AORTA by fulfilling the role of intermediate system (VZVZ, 2013). The central application of LSP is *Zorginformatiemakelaar (ZIM)*, which is a system that contains a reference index regarding patient data in systems of CDOs. This means that ZIM holds data about where patient data is stored on a personal level. It should be explicitly mentioned that ZIM does not store any actual patient data: It only stores the references to the systems that have this data, which makes LSP a *transport facility*. A patient can give a consent to a CDO to register his EMR of that specific CDO at ZIM. At that moment the reference index will be updated with a reference indicating that the system of this CDO holds certain information about this patient. As a result, a physician of any CDO in The Netherlands is then able to request (a part of) the EMR of this person at the CDO via LSP. This is only allowed at the moment that the patient is under treatment of the requesting physician.
Currently LSP only supports the exchange of medication data and *Professionele Samenvatting*, which is a summary of a patient's health record containing information from the last four months (Nictiz, 2012c) and can be requested by general practitioners. Due to this limited support for clinical data, LSP is currently mostly used by general practitioners and pharmacies, while hospital physicians rely on their own EMRs.

Actors

When considering all use cases regarding LSP, VZVZ identifies five types of actors. In order to determine the identity of an actor, several authentication methods are in place. All actors and related authentication methods are stated in Table 3.5 and Table 3.6.

Table 3.5: Actors of LSP (VZVZ, 2013)

Healthcare professional	A healthcare professional is an individual practitioner of a medical profession that can be identified uniquely.
Healthcare provider	A healthcare provider is an organization (or an individual that is not part of an organization) that have a role in the care delivery process.
Employee of a healthcare provider	An employee of a healthcare provider is a person that works for a healthcare provider, but is not a healthcare professional. We can identify an employee uniquely and determine at which healthcare provider he is employed.
Patient	A patient is a uniquely identifiable person of which health data can be exchanged.
Well-managed organization	Well-managed organizations are parties that are not officially entitled as healthcare provider, but have a significant role in the care delivery process.

As stated in Table 3.6, every system that connects to LSP is equipped with a certificate for identification purposes. However, in order to receive such a certificate the system has to be

Table 3.6: Actor authentication methods

UZI card	In the Netherlands, every healthcare professional is registered in the central Beroepen in de Individuele Gezondheidszorg (BIG)-registry, which is maintained by the organization *Centraal Informatiepunt voor Beroepen in de Gezondheidszorg (CIBG)*. Based on this regisry, CIBG issues so-called *Unieke Zorgverleners Identificatie (UZI)* cards, which healthcare professionals can use for online identification (Bonthuis, 2007). This UZI card is provided with a certificate that holds information about the concerned professional, i.e. who he is and what his role is. UZI card identification is the most secure method of authentication that LSP supports.
PKIO-card	Employees of healthcare providers are not registered in the BIG-registry and cannot use UZI card identification for that reason. Therefore the Dutch government has introduced an own *PKI* under the name *Public Key Infrastructure Overheid (PKIO)*. With this infrastructure certificates can be distributed in which the Dutch government is the highest signing authority. This secured certificate infrastructure is used to create unique, personal PKIO-cards for employees of healthcare providers such they can connect to LSP.
UZI server certificate	In order to enable server authentication, CIBG also manages the delivery of UZI server certificates. A UZI server certificate is meant to be able to uniquely identify a system such it can connect to LSP.
Patient	VZVZ describes a patient as a uniquely identifiable person about whom information can be exchanged. The Dutch government has set up the identify provider *DigiD*. This identity provider can be used by all persons, i.e. patients, that have a Dutch BSN (a unique personal number).

accepted as a so-called *well-managed information system* in order to ensure the integrity of the infrastructure.

Interactions

VZVZ describes the functionalities of LSP as so-called *interactions*, which are divided into *primary* and *supporting* interactions. Every piece of data that is exchanged for an interaction is based on the HL7v3 standard. The interactions are stated in Table 3.7 and Table 3.8 respectively.

Table 3.7: Primary interactions of LSP

Handling of requests for patient data
A request that has been placed by a CDO for specific patient data.
Handling of transmission of patient data
When a request is valid, LSP gets the actual data from the target EMR and sends this to the requesting system.

An overview of LSP is provided in Figure 3.2.

Table 3.8: Supporting interactions of LSP

Search for information about CDOs and HISs/EMRs
Before a physician/CDO can send a request for patient data to LSP, it should be known at which CDOs the patient data is available and from which CDO the patient data should be asked.

(Un-)registration of patient data in the reference index
When a patient gives a CDO permission to share his EMR, the CDO will register the EMR at LSP. The CDO will unregister the EMR when the patient withdraws the permission.

(Un-)subscription to a patient
LSP offers the functionality to CDOs to *subscribe* to a patient. This means that the CDO receives a notification whenever new patient data becomes available via LSP.

Configuration of the authorization rules
A patient can give permission to a CDO to allow his EMR to be shared. This authorization is stored in LSP by the CDO or by the patient himself.

Consultation of the audit log
All activity at LSP is saved in an audit log. A patient can access the audit log about his data so he can see whether CDOs have accessed one of his EMRs.

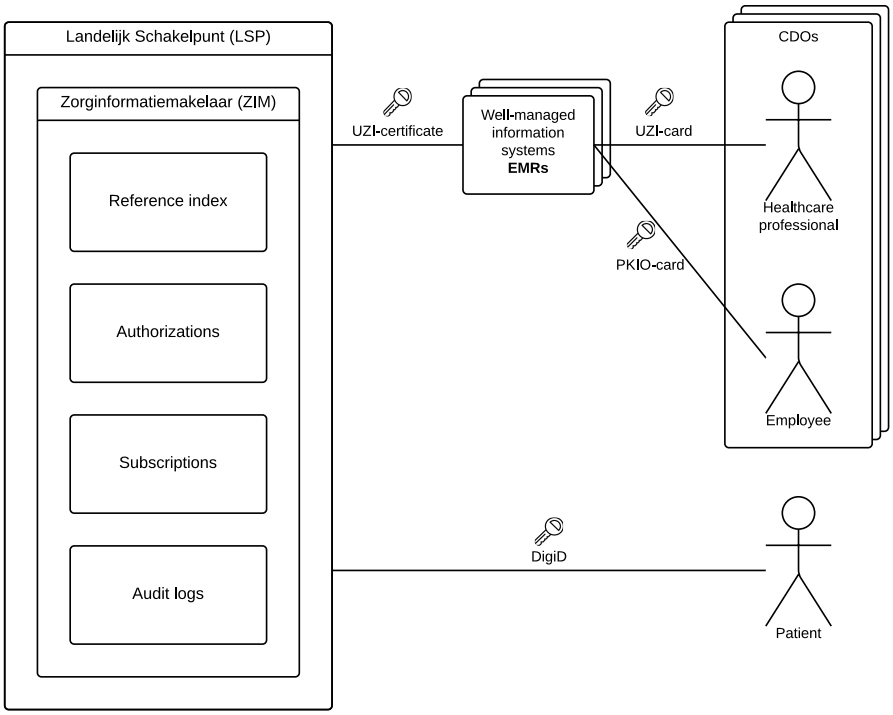


Figure 3.2: LSP overview

3.4 Conclusion

From this analysis we may conclude that in current practice already a lot of developments are in place regarding PHR systems and healthcare interoperability. However currently one PHR

system can be classified as integrated. More concerns arise when we see that only a limited amount of systems are developed such that these can be included in a interoperability scenario, i.e. by having an API. When a PHR system is designed to act as a closed, free-standing system, whether it should be even tried to set up some exchange of health information with such a system. Regarding interoperability standards we may state that HL7 is the leading organization in this field. Although various coding standards, e.g. ICD and LOINC, exist and share overlap, for message standards HL7 standards are practically the most common. Within the Netherlands only a little amount of health information can be exchanged within AORTA. However, the system that is in place to facilitate this exchange, LSP, is relatively advanced and its setup is scalable. Therefore LSP could also fulfill a central role in this research.

Chapter 4

Prototype implementation

In this project we developed a prototype implementation of an integrated PHR. In this section we describe the prototype development by first elaborating on SfH in order to clarify which options exist for setting up an interoperability scenario with the EMR system i.s.h.med. Subsequently we discuss the selected free-standing PHR system that offers functionality to use it in an integration scenario, based on the analysis of PHR systems in section 3.1. Proceeding with a specification of the prototype, this chapter is concluded with a reflection on the prototype by elaborating on the design choices and limitations and providing recommendations for the integration design.
In order to select a free-standing PHR system, the following prerequisites can be set:

1. **Open API**
   We need an open API to have an interface that can be used to read and write data to the PHR system.

2. **Free-standing PHR type**
   We need a PHR system that is free-standing originally. Subsequently we prototype this as an integrated PHR system.

3. **Clinical data support**
   We need a PHR system that supports clinical data, as we are integrating this with a clinical system (an EMR system).

Next to these prerequisites, it is preferred that the PHR system has a community of developers such that we may rely on existing documentation and support. Furthermore it is preferred that the PHR is supported in The Netherlands, as the prototype will be less valuable when a PHR system is selected that is not actually used in The Netherlands.
The PHR system that is used in this prototype is *Microsoft HealthVault*. This system meets the requirements that we have set. Furthermore its platform approach makes that third-party applications, such as our prototype, is part of the design of the PHR system. This is a comfortable situation, as we do not have to "misuse" the free-standing PHR system in order to integrate it with an EMR system. Microsoft HealthVault is detailed in section 4.2.

A video demonstrating the prototype is available at `http://graduation.bastianen.com/`.
A list describing the steps that are required for installing the prototype is given in Appendix G.

4.1 SAP for Healthcare

As explained in section 1.1, SfH is a collection of products for the healthcare domain. As described in the scope of this research (section 1.4), the prototype built in this research will rely on *i.s.h.med* as EMR system. This analysis will elaborate on the architecture of this product and its functionality. Subsequently the interoperability possibilities of this product will be examined, such that it becomes clear how an external system would be able to communicate with i.s.h.med. The core of SfH is ERP Central Component (ECC), which is SAP's ERP system. An ERP system is a modular IT solution that provide general business functionality, i.e. procurement, warehouse management and human resource management (Klaus et al., 2000). ECC is built on the *SAP Netweaver* platform, which provides multiple technological frameworks to integrate and develop SAP applications. ECC can be extended with various additional application layers. One of these layers is *Industry Solutions*, which is relevant for this research. An Industry Solution adds functionality to ECC that is specific to the sector where the system is operating in, in case the standard functionality is insufficiently covering necessary functionality. An overview of SAP ECC's architecture given in Figure E.1. SAP has developed Industry Solution Healthcare (IS–H) to add healthcare-related functionality to ECC. IS–H consists of various packages, e.g. PAB. PAB is the core package of IS–H that adds the typical A-D-T functionality to ECC and enables hospitals to administrate their patients. The combination ECC and PAB can be seen as a HIS, as described in chapter 1.
ECC and PAB are typically sold together with the product i.s.h.med. i.s.h.med covers the package *Clinical Treatment and Care* of IS–H as indicated in Figure E.2 in Appendix E. i.s.h.med is developed by Siemens AG[1] and is fully integrated with SAP PAB. It adds clinical functionality to the existing HIS, e.g. the administration of diagnoses, treatments and medication. Therefore i.s.h.med can be classified as the EMR system for SAP's HIS. Figure E.2 in Appendix E gives an overview of all components of SAP for Healthcare. In Figure E.3 the functionalities of PAB and i.s.h.med are indicated. From this figure the separation of functionality becomes clear.
SAP focuses on developing software that is based on standard, optimized business processes. When this software is delivered, it mainly requires standard configuration activities, so-called *customizing*, which enables the default functionality for this customer. However, PAB and i.s.h.med are highly modifiable and can be fully adapted to the customer's need if wanted. The development strategy of i.s.h.med is significantly different. They believe that CDOs that use i.s.h.med do not have significant amount of commonality among their business processes. Therefore i.s.h.med is delivered as a bare product with a few out-of-the-box features or standard functionality. Instead, i.s.h.med is designed to provide a set of tools to easily create the desired functionality, e.g. data structures and screens, that form the actual EMR system.

Architecture

Currently SAP ECC, in which PAB and i.s.h.med are integrated, is based on a Client-Server model where the server can be hosted on-premise or in the cloud. This Client-Server model is designed as a three-tier architecture (Schuldt, 2009) for the basics of SAP ECC in which it is split in a *data*, *application* and *presentation* layer. The technologies used in the configuration of a client or server can be different per installation, i.e. SAP ECC can be installed in dozens of different configuration setups regarding the choice for operating system, database engine, hardware architecture etcetera (SAP, 2014). By porting the SAP Netweaver platform to every supported configuration, the SAP ECC application remains the same and does not require different implementations for different configurations.

[1] In August 2014, i.s.h.med is acquired by the corporation Cerner.

Interoperability

SAP ECC offers a wide range of options to interoperate with an external system. Depending on the requirements for an interoperability scenario, decisions can be made towards the realization of this scenario. In Table H.1 in Appendix H various techniques are discussed to set up an interoperability scenario with SAP ECC. Together with online documentation (SAP SE, 2014a), a schematic overview of interoperability has been composed as showed in Figure 4.1. In Table 4.1 the interoperability options are listed and subsequently categorized according to the methodology of Hohpe & Woolf (2004), as described in section 2.7.
Based on a discussion with a field expert, as described in Table H.1 in Appendix H, the preferred set-up for an integration scenario with SAP ECC is to use SAP Process Integration (SAP PI) as *integration broker* in combination with Advanced Business Application Programming (ABAP) Proxies. This enables you to have an organized, clear setup with a single point from where inbound and outbound messages are routed to their destinations while the separation of the logical components of your IT landscape remains.

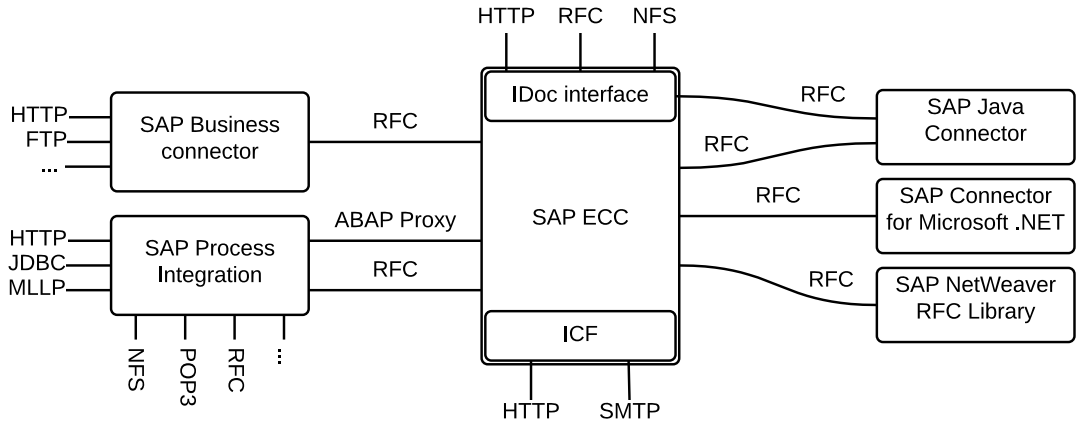


Figure 4.1: SAP ECC interoperability options

Table 4.1: SAP ECC interoperability options

		File Transfer	Shared Database	Remote Procedure Invocation	Messaging
SAP BC	SAP Business Connector (SAP BC) is a so-called *integration broker*, providing *middleware* functionality. It supports various protocols, message types and has the ability to translate messages to other formats.				X
ICF	Internet Communication Framework (ICF) is an integrated component of ECC that is able to expose internet-enabled interfaces to external applications.	X			
IDoc interface	The Intermediate Document (IDoc) interface is an integrated component ECC that is capable of processing inbound and outbound IDocs over HTTP, RFC and NFS. An IDoc is a standardized SAP XML-document describing a business transaction, e.g. *create sales order*.	X			
SAP JCo	SAP Java Connector (SAP JCo) is a SDK for Java-applications that embeds libraries for connecting to ECC via RFC. Furthermore it is capable of accessing the IDoc interface of ECC for transferring IDocs.	X		X	
SAP NCo	SAP Connector for Microsoft .NET (SAP NCo) is a SDK for .NET-applications that embeds libraries for connecting to ECC via RFC.			X	
SAP NRL	SAP Netweaver RFC Library (SAP NRL) is a SDK for C and C++-applications that embeds libraries for connecting to ECC via RFC.			X	
SAP PI	SAP Process Integration (SAP PI) is the successor of SAP BC that functions as middleware component in an integration scenario. It is equipped with different *adapters* that are capable of processing messages in different formats and over a long list of protocols.				X

4.2 Microsoft HealthVault

Microsoft HealthVault is a free PHR system developed by Microsoft where anyone can create an account and subsequently use the PHR system. A HealthVault user has one or more PHRs in which he can enters data via a web interface or mobile application. However this web interface is limited to standard import/export functionality and manual entry of user data. The distinctiveness of HealthVault, compared to other PHR systems, is that it is deployed as a platform rather than as an application. This means that HealthVault provides a base for developers to create applications that use HealthVault as underlying platform or to connect devices directly to the platform, i.e. a smart scale. HealthVault applications are able to provide more advanced functionality regarding data processing, such that HealthVault manages for example user accounts, user data and authorizations while the application is able to interface with the platform and provide actual real functionality to the user. In order to help the development of HealthVault applications, Microsoft has equipped HealthVault with an extensive API, based on XML Web Services. This API can be used in two ways: Either in a raw, direct way or developers can make use of the SDK that embeds these web services in Dynamic-Link Libraries (DLLs) to be used in .NET-application development projects. A list of the methods that are supported by the API is given in Table F.1 in Appendix F. These web services and web service methods are tailored for the HealthVault platform.
In order to indicate the variety of data types that HealthVault supports, all data types are listed in Table F.2 in Appendix F. In the list of data types, a few interoperability standards can be found. The HealthVault platform currently has native support for CCD and CCR formats, which are developed for exchanging complete health records. CCD is the HL7 implementation of the earlier existing CCR standard. However, due to the fact that these standards are developed for the exchange of complete health records, these standards require a holistic approach as confirmed by Wagner & Jones (2009). This means that when interpreting a CCD or CCR document, it should be treated as a *snapshot* of complete record and should be interpreted as a whole. Depending on what information needs to be exchanged, CCD or CCR can be an appropriate format. However, when you need to exchange only a little piece of data, e.g. a vital sign measurement, it can be doubted whether CCD or CCR is the right format to do so, because it is not designed for that purpose. This trade-off also becomes visible when observing the implementation of CCD and CCR. When such a document is uploaded to a PHR, the segments of the document require so-called *reconciliation*, which is the manual process of indicating for every segment whether it should be saved in the PHR.
Due to the privacy-related laws and legislation, e.g. the Patriot Act, in the United States of America (USA), non-American citizens prefer to not store any personal data at a party that is subject to these laws and legislation. Microsoft has solved this issue by running multiple instances of HealthVault in different countries: An instance in the USA and an instance in the United Kingdom (UK). Depending on the country that a HealthVault user lives in, the user contracts a subsidiary of Microsoft when he creates a HealthVault account, as described in clause 15 of the HealthVault Service Agreement (Wagner & Jones, 2009). Next to these two instances of HealthVault, Microsoft also runs a Pre-Production Environment (PPE) in the USA and in the UK, such that developers have a platform to develop and test their applications.

Authentication and authorization

Due to the fact that HealthVault applications are not hosted by Microsoft but are hosted at third parties, a mechanism is required to unambiguously identify an application. Every HealthVault

application is therefore uniquely registered at HealthVault with an application identifier and provides an application certificate for an application. This certificate is used by the application to authenticate itself when connecting to the HealthVault platform, such that the HealthVault platform can verify the identity of the application and the connection integrity is ensured. When registering an application with HealthVault, the developer defines the access that the application requires to a PHR in order function properly. These definitions can also be marked as optional, in order to enable the PHR user to choose whether the application is allowed to access this data type (and may receive additional functionality from the application) or to deny access. These access definitions are called *rules*. A rule is a combination of a data type and an access type. The access types are the basic data operations Create, Read, Update and Delete (CRUD). HealthVault distinguishes *online* and *offline* authorizations, such that a developer can define the required access when the concerned PHR user is online and when he is offline separately. HealthVault offers multiple options for applications to connect to a HealthVault PHR, in order to enable applications to achieve the required application behavior. The options differ mainly on how or when the user authorizes an application to connect to his PHR. The options are listed in Table 4.2.

Table 4.2: Authorization-based access options (Microsoft, 2015a)

Web-based authorization	In case the application is equipped with a website that the user can access, authorization can be provided by the user by visiting this website. The application website will redirect the user to HealthVault where he has to log in to HealthVault. The next step is to authorize the application based on the required access rules that are shown and will be redirected back to the application subsequently. In case the application requires offline access to the PHR, the application stores the connection details such that it can reuse the authorization.
Patient Connect	In case the application is not equipped with a website, HealthVault provides an asynchronous method for application authorization. In this case a representative of the application asks the user to provide a secret question and answer, which the representative will submit to the HealthVault platform. HealthVault will respond with an unique code that will be handed over to the user. Later the user will visit a special web page of HealthVault where he can enter the unique code. Subsequently the user is prompted with the secret question and is required to give the answer that the representative has submitted earlier. The last step is to authorize the application to access the PHR based on the access rules that are shown to the user. In parallel, the concerned application constantly checks with a polling job whether there are new users that have authorized the application to access their PHRs.
SODA	Software on Device Authentication (SODA) is an access option tailored for applications for devices, e.g. mobile phones and tablets, where the application is installed locally. In case this application is set up to have a direct connection to HealthVault without routing via a server of the application, HealthVault should be able to uniquely identify a specific instance of the application as the user wants to specifically authorize this instance to access his PHR. This is achieved by the fact that the developer registers a so-called *Mobile master application* in HealthVault. This master application acts as the parent of every instance, such that it inherits all attributes. Subsequently when a specific instance is opened for the first time, it reports its existence to the HealthVault platform by calling a dedicated HealthVault web service providing the identifier of the master application. HealthVault then registers the instance as a child of the master application and the user is able to authorize the instance subsequently.

4.3 Specification

For the prototype of an integrated PHR an asynchronous integration scenario has been set up between i.s.h.med and Microsoft HealthVault. As the scope of this research describes in section 1.4, the prototype will cover a delimited, logical scenario. Therefore we searched for a useful scenario to implement, such that the prototype reflects a limited, but real-life scenario. We choose to consider the field of bariatrics[2], as multiple sources indicate that PHRs are primarily valuable for people suffering from a chronic disease (Archer et al., 2011; Detmer et al., 2008; Dorr et al., 2007). Based on this disease, the prototype has been setup to support the exchange of weight measurements, which is one of the important measures for bariatric patients.
In order to increase the value of the prototype, we have applied best practices, preferred design principles and standard procedures of SAP NL and Microsoft HealthVault. This results in the following prototype characteristics:

1. Data exchange with i.s.h.med is set up over SAP PI combined with ABAP Proxies, which is the preferred setup for messaging of SAP.

2. The prototype incorporates the HealthVault access option *Patient Connect* as the registration process of patients.

3. The prototype incorporates an HL7 messaging standard as a message format for the data exchange.

4. The weight measurements are stored in the default location for vital signs within i.s.h.med.

In Figure 4.2 a schematic overview of the prototype architecture is shown. In order to be able to interoperate with HealthVault, we built an application called *HealthVault Integrator* using the HealthVault .NET SDK provided by Microsoft. As shown the HealthVault Integrator consists of two components. The component *Patient manager* is meant for the registration of patients and *Scheduler* is responsible for the actual exchange of data with the HealthVault platform. As the HealthVault Integrator should be able to interface with the EMR system, in practice the application will be running in the Local Area Network (LAN) of a CDO. However, the HealthVault Integrator should also be able to interface with the HealthVault platform, which means that it should be connected to the internet. This setup can be potentially bring security vulnerabilities as there is a direct, internet-enabled interface to the EMR system. Therefore we have included a "decoupling" point in the architecture by storing the HL7 message, that has to be transmitted between SAP PI and the HealthVault Integrator on the file system.

HL7 conformance

The HL7 message that is used for transmission of weight measurements, is the message standard *ORUˆR01* version 2.5.1, which is the most recent version of this message standard in the HL7v2 standard. ORU messages are developed for the transmissions of unsolicited results, e.g. a measurement done by some machine. For height and weight measurements HL7 prescribes the use of an unsolicited[3] ORUˆR01 message (Madra et al., 2013). According to this implementation guide, the HealthVault Integrator that we have built is fully compliant to the standard. For the actual parsing and encoding of the messages the open source project *nHapi* (Edwards, 2014) has been implemented in the HealthVault Integrator. nHapi is a port of the Java project *Hapi* to

[2] Bariatrics is the medical specialty treating obesity.
[3] *Unsolicited* indicates that this result message is not preceded by a request message. This behavior is often found with lab tests that are first requested and subsequently "answered" with a result message.

the Microsoft .NET-platform. nHapi provides the full object model of HL7v2.x and is capable of parsing and encoding these messages into a pipe-delimited or XML format. In this prototype XML formatting is used of which an example is stored in Listing G.1 in Appendix G.
In case we would have used another message standard, e.g. based on HL7v3 or FHIR, the setup would not be very different. Only the HL7 message, as illustrated in Figure 4.2 would have been different. However, in case FHIR was selected as standard and we choose to include the implementation of the RESTful API, the applications SAP PI and the HealthVault Integrator should have been equipped with these interfaces, or at least those that are required for the prototype implementation of weight measurements. We could can directly send the messages over these web services.

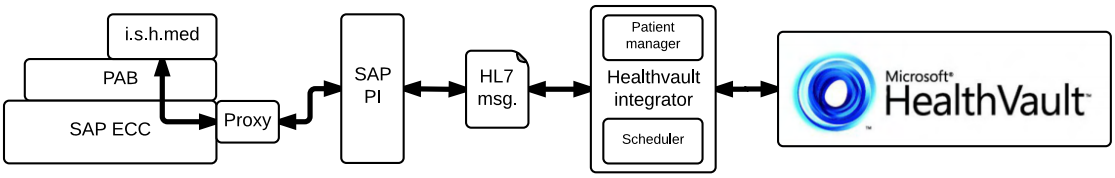


Figure 4.2: Prototype architecture

Use cases

The full integration scenario can be structured into five use cases. These use case are discussed separately in the remainder of this section. Furthermore each use case is supported by a sequence diagram, stated in Appendix G. In general, a sequence diagram is used to show the interactions between objects (IBM, 2004). In this case, the diagrams are used to show a pseudo-style, graphical representation of the code of the prototype in order to clarify the decisions and checks that are carried out by the different systems and components in the prototype architecture. These systems and components are explained in Table G.1 in Appendix G. The sequence diagrams do not provide a complete specification of the prototype, but are focused on the primary sequence. For example input validation and error handling are intentionally left out of the diagrams to prevent complexity.

Register patient

The component *Patient manager* is a web application that contains the functionality to establish a link between an EMR and a PHR by registering patients and subsequently triggering the process for the Patient Connect process of HealthVault, as described in Table 4.2. This process starts for example by a patient asking for an integration between his HealthVault PHR and the CDO's EMR, which is evaluated by a clerk of the CDO. When the request is accepted, the clerk uses the HealthVault Integrator to send an integration request to HealthVault. At that moment the Patient Connect process has been initiated. A schematic overview of this process is stated in Figure G.1.

The sequence diagram related to this use case is stated in Figure G.2.

Create HL7 message from i.s.h.med observation

This use case is *event-based* as we trigger the creation of an outbound message at the moment that a new weight measurement is stored in i.s.h.med. This is done by implementing the Business Add In (BAdI) *N1_VITPAR_DIALOG*. A BAdI is a predefined point in a standard ABAP program where a SAP customer can add custom code. This BAdI is called when saving a vital sign in i.s.h.med and checks the type of the saved vital sign. If the type is *weight*, the BAdI determines whether the saved vital sign is a new or an updated weight measurement. Subsequently a custom function module is invoked that fills the message structure of the HL7 message ORUˆR01. This structure is then encoded to an XML file and pushed the outbound ABAP proxy. The proxy then triggers the *integration flow* in SAP PI that will save the XML file in the watch folder on the file system.

The sequence diagram related to this use case is stated in Figure G.3.

Send HL7 message to HealthVault

This use case is based on a polling operation. On a frequent base, i.e. every minute, the HealthVault Integrator checks whether HL7 messages, which have to be send to HealthVault, are present in the watch folder. For each message found, the message structure will be validated and subsequently decoded. Based on the patient identifier in PAB that is present in the message, the HealthVault Integrator checks whether the patient is registered for an integration with Microsoft HealthVault. The value of the element *OBX.11* in the HL7 message indicates whether the weight measurement is new ('F') or it is an update of an existing measurement ('C'). Next, the HealthVault Integrator creates a connection request to HealthVault. In case HealthVault allows access to the PHR of the requested patient, it will return a one-time connection key that is then used to store the data in the HealthVault PHR. Concluding the HL7 message is archived.

The sequence diagram related to this use case is stated in Figure G.4.

Create HL7 message from HealthVault observation

This use case is based on a polling operation. On a frequent base, i.e. every minute, the HealthVault Integrator checks for all registered patients whether new or updated weight measurements are available. This process starts with updating the registered patients by checking whether there are new or updated patient registrations for the application. Subsequently for each patient a connection request is created in order to access the related PHR. The application then updates the locally stored personal information of the patient, followed by a call for all new or updated weight measurements. In case one or more measurements are retrieved for a patient, an ORUˆR01 message is encoded and saved in the watch folder.

The sequence diagram related to this use case is stated in Figure G.5.

Send HL7 message to i.s.h.med

This use case is based on a polling operation. On a frequent base, i.e. every minute, SAP PI checks whether HL7 messages, which have to be stored in i.s.h.med, are present in the watch folder. For each message found, the message structure will be validated and is subsequently pushed to the inbound ABAP proxy. Within the proxy the message is decoded. Each weight measurement that is present in the message is then processed. Based on the value of the element

OBX.11 in the HL7 message, a new measurement is created or an existing measurement is updated. Concluding the HL7 message is archived.

The sequence diagram related to this use case is stated in Figure G.6.

4.4 Reflection

During the development of the prototype numerous design choices have been made in a functional and technical perspective. Some choices are justified by the fact that we are developing a prototype, other decisions have been made based on what is actually better in the context of an integrated PHR. Furthermore limitations may arise as external applications, e.g. i.s.h.med, SAP PI and Microsoft HealthVault, are used that offer a predefined set of functionality. In this section we reflect on the design choices and limitations that arose during the prototype development are discussed and substantiated.

Architecture

In the initial phase of the prototype development, decisions had to be made upon its architecture. Either the functionality of SAP PAB and i.s.h.med could be extended such that the XML API of HealthVault is directly accessed from i.s.h.med, optionally via SAP PI and ABAP proxies, or an intermediate application is created that will be connecting to HealthVault's API. We choose to implement the second option, the *HealthVault Integrator* as it will enable us to use an interoperability standard, i.e. HL7v2, rather than dedicating functionality to the HealthVault platform. Furthermore this application can be reused to enable other clinical applications (other EMRs) to connect to the HealthVault PHR, because the application is independent from SAP's products. The HealthVault Integrator can therefore be considered as a minimalistic *messaging bus*.
A disadvantage of the selected setup is that neither i.s.h.med does not have information about which patients have enabled HealthVault integration. Therefore i.s.h.med is unable to determine for which patients it should create an outbound HL7 message and for which patients it should not. This issue is resolved in the prototype by creating an outbound message for every weight measurement that is saved. The HealthVault Integrator is subsequently evaluating whether the message required further processing or should be ignored based on the registered patients. Clearly this functionality is undesirable as it exposes sensitive patient information to an application outside of the EMR. Although this information will remain inside the LAN of the CDO, it is preferred to only extract data from the EMR that is allowed to be exchanged.

Message format

For the exchange of data, a message format should be chosen that is mutually, i.e. the sender and the receiver, agreed upon. This message format can be custom and fully tailored to its purpose. However, this would require a transformation of data at both the side of i.s.h.med and HealthVault. The second option would be to take the data structure of a weight measurement of one of the systems, i.e. i.s.h.med and HealthVault, and extend that to a message format for data exchange. The advantage of these two options is that there will be only little overhead in terms of the information that is included in the message as it will only contain the minimal subset of the information that needs to be exchanged. However, such a message format will clearly not stimulate the interoperability and openness of this prototype. We choose to select a message format from the HL7v2 standard as this is currently the most widely used standard within the

healthcare domain. Although HealthVault natively supports CCD messages, this message is not appropriate for the exchange of weight measurements, as explained in section 4.2. Therefore we selected the HL7v2.5.1 that is intended for the use of exchanging weight measurements.
The implementation guide of the HL7v2.5.1 ORUˆR01 message for weight measurements prescribes almost completely how to implement it. The guide only leaves open the way that the fields *MSH.3*, *MSH.4*, *MSH.5*, *MSH.6* are filled as these segments contain the identifier and description of the sender system and the intended receiver system of the message. All identifiers that are used by the network of parties that exchange data via these messages should have agreed upon these identifiers in order to understand which identifier belongs to which system of which party.
During the implementation of the HL7v2.5.1 ORUˆR01 message, we faced a technical limitation found in the system SAP PI. According to the message standard the element *OBX.5*, which contains the observation value, has the type *varies*, which indicates that the type of the value that will be entered in this element is dependent on the value in another element (in this case *OBX.3*). However SAP PI does not support elements that have this field type. Therefore we have adapted the XML Schema Definition (XSD) for this element and changed the type to *numeric*.
Another limitation can be found in the definition of the HL7v2.5.1 ORUˆR01 message as there is no support for communicating the deletion of a measurement. An observation value status is constraint to the values of the standard HL7 Table *0085*, which only accounts for *corrective* and *final* results.

Triggering

As described in the use cases (section 4.3) we have intentionally implemented two methods for triggering a process, such that the differences and related limitations of each method become clear. The first method is *event-based* triggering, which means that the process is initiated at the moment that an event has occurred, e.g. saving a weight measurement in i.s.h.med (use case *Create HL7 message from i.s.h.med observation*). Another way of triggering is to have a *polling operation* that checks whether a process should be started. Polling is the operation of performing certain checks on a predefined frequency or schedule, e.g. checking whether there are new or updated records available in a PHR (use case *Create HL7 message from HealthVault observation*). Event-based triggering and polling can be compared to respectively push and pull types of communication.
Event-based triggering is known to be a more efficient way for an interoperability scenario as it will only cause communication to take place when there is actual data to exchange, while polling operations can take place many times with the result that there is no data to exchange. Another possible disadvantage of a polling operation is that in case some data is changed more than one time in between of two polling operations, these updates are not exchanged. On the other hand, when you are dealing with an external application, in practice it is often less complex to implement a polling operation that checks whether new data is available than creating a way to make the external application notify you about the occurrence of an event.
For this prototype event-based triggering shows to be a more appropriate triggering method as polling is too limited. The current prototype does not support the deletion of records in HealthVault due to the fact that data is retrieved from HealthVault by a polling operation that is subsequently transformed to an HL7 message, which is then asynchronously processed by another polling operation (SAP PI). These difficulties are resolved when the creation process of a message is triggered event-based, which can be realized by using the Event Notification Service (Microsoft, 2015b). This service can be managed via the XML API or the SDK of HealthVault. A HealthVault Application can register so-called *event subscriptions*. Such a subscription contains

a data type, a CRUD-operation and the delivery channel for the notification, which has to be a Hypertext Transfer Protocol Secure (HTTPS) destination. We have seen similar functionality at the subscriptions feature of AORTA, described in section 3.3.

Complexity

In this prototype a relatively simple integration scenario has been implemented in terms of the size and the dimensions: One EMR integrated with one PHR in which a single-attribute data object is exchanged that has a relatively static nature, i.e. the value is not subject to many changes, and can only be changed by its source. A slight modification or extension of the context of the prototype could have a significant impact on the current implementation as the current implementation does not sufficiently foresee scalability. When the context is expanded to more real and complex situations, the following topics will have to be addressed to ensure a properly integration scenario:

Multiple systems In the current scenario one EMR and one PHR are considered. In case we expand this to for example two EMRs, the implementation should ensure that data gets send to both EMRs and that possible changes are propagated to these systems as well.

Multi-attribute objects In the current prototype a single-attribute data object is handled, which is a relatively simple situation. In case of a multi-attribute object, such as an object describing an immunization or a lab result, the data object will become more complex. Depending on the level of maturity of existing interoperability standards such data can be exchanged properly. An even more complex situation would be if the interpretation of data is depending on multiple data objects. This would require a specific set of data objects to be able to properly evaluate the situation. This situation can for example occur when side effects of certain medication is evaluated through lab tests and vital sign measurements.

Multiple editors Combining the situations of having more than one system next to the PHR system and having a multi-attribute data object, the integration scenario would become much more complex when every party is allowed to update any piece of information. When drawing a possible example, it may happen that when one physician changes the dose of a medication that another physician, from another CDO, has prescribed to the patient, while later the pharmacy substitutes this medication for a comparable medicine from another brand. Clearly it becomes a complex situation when multiple parties are managing the same object. This complexity concerns various topics, e.g. data propagation and integrity, responsibility/ownership, governance and patient consent. When allowing this complexity into an integration scenario, the impact should be properly investigated regarding all these topics.

Multiple associations Similar to the *multiple editors* complexity, we may think of delimited pieces of health information that are by default associated with multiple parties, e.g. medication is typically not prescribed and delivered by the same person. This causes complexity in terms of responsibility/ownership, data propagation and integrity.

4.5 Recommendations

The prototype implementation has provided valuable insights and focus areas regarding the implications of integration. Based on the findings that are described in section 4.4, the following recommendations can be defined towards the integration design that is explained in chapter 5.

Consent awareness at source system Any source system should be able to determine which integration scenarios are active for a patient by checking which type of consent a patient has given for some integration scenario and which data is involved with this.

Interoperability standards Clearly an integration scenario with external parties requires an interoperability standard. In order to improve the scalability of the scenario, it is preferable to select an existing standard. Considering these, the concept of HL7 FHIR would be the preferred choice. However, it could be that parties prefer to not use a standard that is DSTU. In that case HL7v3 would be the preferred choice, due to the fact that it has an underlying information model.

Event-driven messaging In order to ensure data integrity, source systems should actively notify target systems of new or updated information. Therefore new or updated health information that is part of an integration scenario, should be reported to the target systems in an event-driven approach. Furthermore an event-driven is a more efficient way of communication between two systems as communication will only take place at an event, which indicates that an actual message is ready for transmission.

Scenario integrity Integration scenarios can easily become relatively complex as explained in section 4.4. Due to the nature of health data, topics like sensitivity and integrity are present by default. Therefore any integration scenario has to be developed such that there is a minimized, manageable risk of errors by design, while complying with the applicable laws, legislation, patient consent etcetera.

Chapter 5

Integration design

In this chapter we present the design for a generic integrated PHR scenario that is independent of the EMR system, the PHR system and the number of systems involved. We start with listing the actors that have a relevant role in the setup of an integrated PHR scenario in section 5.1, such that we can uniformly refer to them in the remainder of the design. Subsequently we summarize the challenges that can be identified for the setup of an integrated PHR, based on additional literature review and the information that has been gathered in the initial literature review, analysis and prototyping phases of this research (section 5.2). By clustering these challenges into more general topics, we are able to establish a structure for the design of the integrated PHR scenario. Based on the clustered challenges we will describe the design, which we structure by defining *design principles*. The design principles will be the formal input for the architecture that is provided in section 5.3. Concluding we present a road map that indicates which steps should be taken in order to realize an integrated PHR scenario.

5.1 Actors

In this section the actors are defined that fulfill a role in the scenario of an integrated PHR.

Table 5.1: Actors in the integration design

Individual	The *individual* refers to the single person about who we can exchange data and is the central subject in this integrated PHR scenario.
Participant	The term *participant* refers to any entity, e.g. an individual, a CDO or a physician, that can exchange data about the individual.
Neutral party	The *neutral party* refers to the corporation that is responsible for choosing and extending interoperability standards.
EMR system	The *EMR system* refers to the system that is used by a CDO.
PHR system	The *PHR system* refers to the system that is used by the individual.

5.2 Design

It is clear that introducing an integrated PHR is a complex process. Numerous barriers can be found in literature or can be derived from the development of our prototype. Tang et al. (2006)

studied the barriers for the adoption of PHR systems. They found the following barriers for the realization of an integrated prototype:

- The PHR system should support the same interoperability standards as other health information systems.
- The individual should be able to specifically allow certain segments of his PHR to be shared.
- A mechanism should be in place to trustfully authenticate the individual.
- The data that has been entered by a patient can be (considered as) inaccurate data.

Based on a survey by the Raad voor de Volksgezondheid en Zorg (RVZ), the Dutch council for public health and healthcare, the biggest concern of individuals is the risk of privacy violation in case their health-related data is accessed by people against their consent (Raad voor de Volksgezondheid en Zorg, 2014). Furthermore Detmer et al. (2008) investigated the potential of integrated PHRs in the USA and identified the following issues:

- A lack of incentives for physicians to promote an integrated PHR due to a shift in patient autonomy, changing physician responsibilities and concerns about liability risks.
- A lack of the individual's confidence and trust due to security and confidentiality concerns.
- A lack of technical standards for interoperability regarding data interchange standards, authentication processes, security standards, certification etcetera.
- A lack of health infrastructure due to the high costs that are involved when integrating data that originates from various systems and the absence of a mediating structure.
- "The Digital Devide", describing the division of individuals that have and have not the ability to use digital IT solutions.
- An unclear value realization and the return on investment due to the fact that it is unclear to quantify the value proposition of an integrated PHR and due to the potential barriers and hurdles for realizing an integrated PHR system.
- An uncertain market demand due to the fact that integrated PHRs are a new phenomenon and the consequential lack of information about how it is going to be used.

Furthermore Detmer et al. (2008) list the following issues as top priorities when it comes to what an individual demands from a health information exchange network:

- The prevention of unauthorized access of any case of mistaken identity.
- The ability for the individual to review who has had access to his health information.
- The need for an individual's consent to share his data with other participants.
- The prevention of employers and insurance companies of accessing a health information exchange network.

Summarizing within the scope of this research, the key challenges for the implementation of an integrated PHR are concerning the topics authorization, authentication and interoperability. These topics are discussed separately in the remainder of this section.

Authorization

Clearly an individual should be and wants to be in control of the exchange of his health information. This will require a PHR system to have an authorization model in place in which an individual can manage the access to his PHR for participants. A study of Carrión Señor et al. (2012) shows that in the majority of available PHR systems the individual may control who can access their PHR. However, a much smaller amount also offered the individual an audit log indicating who had accessed the PHR and when. Rø stad (2008) discusses an initial model for a role-based access control in PHR systems in which three topics are highlighted: Simplicity, time and transparency. An individual should be able to reuse predefined roles or to the define own roles that have certain permissions in the PHR. Subsequently the individual can assign roles to a participant based on time periods in order to give consent for data exchange. Concluding the access control mechanism provides transparency by offering tools for reviewing audit logs. Based on this model and the analysis of authorization methods that have been evaluated for the HealthVault platform, we come to the following authorization design principles:

> **Principle 1.** *The individual can define which participant is allowed to access his PHR (a consent).*

> **Principle 2.** *The individual can refine a consent such that the participant can access only a part of his PHR.*

> **Principle 3.** *The individual has the ability to revoke his consent.*

> **Principle 4.** *The individual can review audit logs, such that he can verify whether no unauthorized access has been provided to his PHR.*

As described in section 3.1, many PHR systems provide an emergency access method to a PHR. This can be a valuable feature of a PHR system. However, especially in case of emergency access it is unnecessary to get full access to an individual's PHR. This leads to the following design principles:

> **Principle 5.** *The individual can define which parts of his PHR can be accessed when accessing the PHR with an emergency access key.*

Authentication

Mistaken identities or even identity theft can occur when an actor is not authenticated properly. The simple mechanism of a user name-password combination is almost an obsolete authentication method nowadays. In the scenario of an integrated PHR, where individuals provide specific consents to specific participants or systems, it is important that the used authentication methods are reliable and strong. For human actors in this context the use of a multi-factor authentication method is inevitable when considering the sensitivity and confidentiality of the health information that is exchanged.
Within The Netherlands every actor is capable of identifying itself via a strong authentication method, as described in Table 3.6 in section 3.3. The big advantage of the Dutch setup is that an

identity provider is used for handling the authentication. The advantage of an identity provider is that the *identities* can be reused[1].

Principle 6. *Any actor is uniquely identifiable in order to ensure the authenticity of this actor and to be able to perform proper auditing.*

Principle 7. *Any actor uses an authentication method that is considered to be sufficiently reliable and strong.*

Interoperability

In the prototype health data is synchronized between a PHR system and an EMR system and is stored in the systems separately. This approach is acceptable when the integration scenario involves only a few EMRs, but gets complex in case multiple EMRs are involved or even multiple PHRs. Although it is unlikely that an individual will use multiple PHR systems, we see that WHMSs typically store their measurements in an included, dedicated PHR system. The scenario becomes even more complex when data is modified in some system and has to be propagated again to other systems. In the worst case data at some system gets outdated, because the propagation fails or because the individual has revoked his consent for this participant/system in the meantime.
A solution to this problem is to build a central data repository, an EHR, where data is replicated from one system to a central data store. Subsequently other systems may use this central repository to retrieve data. This centralized approach gives various opportunities and advantages. Apart from the fact that the data propagation and scalability issues are resolved, a central repository simplifies the integration and retrieval of data. However a central repository brings also various disadvantages which are mainly security based. The fact that one data repository contains the sensitive health data of a big number of people raises a lot of privacy concerns (Bergmann et al., 2006). These issues indicate that it is expected that the same concerns arise when an individual's health data is stored in a single PHR system. These expectations are aligned to the questionnaire results of Raad voor de Volksgezondheid en Zorg (2014) that found that more than 75% of the respondents think that it is not a good idea that a party like Microsoft or Google takes care of storing health data.
A third option for establishing an integration scenario is to set up virtual links in stead of physically propagating the data across the participants, which is similar to the architecture of the Dutch infrastructure AORTA, as explain in section 3.3. This interoperability method relies on a central index that keeps track of where certain data is available rather than storing health data centrally. Furthermore it resolves any data propagation issue as no data is transferred other than when the information is explicitly requested, which ensures that the data is always up-to-date. Also privacy and security concerns are less critical as data remains at the point of care. The disadvantage of virtual links is clearly that it is a relatively complex integration scenario, but as it resolves the earlier found issues and a similar setup is already successfully in place in The Netherlands, the following design principle is defined.

[1] A well-known international identity provider is *Facebook* that offers *Facebook Connect* as an authentication method. A well-known Dutch identity provider is *DigiD*, which is used by citizens for logging on to government websites.

Principle 8. *The integration of data is achieved by setting up a virtual link to the EMRs and PHRs.*

When the individual opens his PHR, he will notice all information that is available from any EMR, while the virtual links make that this PHR "physically" only contains the health data that has been created by the individual himself. However, when an individual wants to store his data locally in a PHR he should be able to do so.

Principle 9. *The individual may still choose to store data from an EMR in his PHR.*

As data is being exchanged, which originates from multiple health records, we need a mechanism that ensures that every piece of information can be identified uniquely across the infrastructure. Especially when data that is already exchanged gets updated, it should be able to unambiguously identify the original data that needs to be updated in a target system.

Principle 10. *Any piece of information that originates from a health record is uniquely identifiable.*

Clearly this virtual integration scenario has a strong need for interoperability standards. Primarily because data should be transferred in a jointly agreed format, but also because systems should be able to notify the *central index* about which information they hold about some individual. Therefore every involved system in the integration scenario has to comply with a predefined standard, which is preferably a recognized standard. This standard should be selected by a neutral party that has in-depth knowledge on the domains of interoperability and healthcare, e.g. Nictiz[2] in The Netherlands. One important note to this is that the use of standards should not be a limiting factor. This limitation is twofold: On the one hand, the standard should not be that complex or hard/expensive to implement that smaller CDOs, EMR system developers or PHR system developers are not able to join this integration scenario. On the other hand, the standard should be sufficiently open and broad, such that most health information can be exchanged and not only a subset of this. It may happen that the chosen standard does not provide one hundred percent support for the data that parties want to exchange, e.g. in case of country specific data. This neutral party should then be able to define and release extensions on the chosen standard.

Principle 11. *A neutral party should be in place that decides on which standards are used and how these are optionally extended.*

Principle 12. *Any message that is exchanged complies with a predefined interoperability standard, set by the neutral party.*

[2] "In consultation with and at the request of the healthcare sector, Nictiz develops and refines national standards for electronic communications in healthcare." (Nictiz, 2014b)

Principle 13. *The interoperability standard should not be limited in the sense that participants could be excluded due to the too big effort to support the standard.*

The virtual links approach prevents a participant from modifying data that is not originally created by him. The same holds for the individual who is not able to modify data in an EMR. As described in section 4.4, when discussing *multiple editors*, it is stated that reasons exists for supporting data objects to be modified by multiple actors. This can be realized when *change requests* are included in the integration scenario. Such a request will be send to the actor that originally created the record and will be evaluated subsequently. At the moment that the change is honored and processed, this will be automatically notified at the central index. Note that such governance could be potentially automated under certain conditions such that it does not cause unnecessary overhead.

Principle 14. *Any data object can only be modified by its creator. Other participants need to request a change.*

5.3 Architecture

In this section we propose an architecture based on the design principles that are provided in section 5.2. Although this is done in a general sense, we will detail the architecture towards the current practice in The Netherlands regarding health-related data exchange. We motivate this from the fact that laws, current infrastructures and processes in this field have a strong country-dependent or even region-specific orientation. Furthermore we motivate this because AORTA has some similarities compared to the approach that is described by the design principles. In Figure 5.1 a renewed architecture overview is shown.

Authorization Authorizations are centrally maintained in LSP. An individual can connect to LSP directly, as in the current version of AORTA, to give his consent and to consult the audit log. However, the types of consent that the individual can store in this integration design are on a more refined, detailed level. Every participant will have to start interacting with this *authorizations* module of LSP in order to check which information an individual wants to be exchanged, based on the consents that are registered.

Authentication The authentication methods, i.e. Unieke Zorgverleners Identificatie (UZI) and DigiD, will remain as prescribed by the Dutch *Code of Conduct for health information exchange* (Hodes, 2013). Additional efforts are required for PHR systems in order to support the required authentication method for individuals, i.e. DigiD. As this is a Dutch identity provider, this requirement could potentially narrow the range of PHR systems that are supported for this integration scenario. In case the PHR system is developed by a non-Dutch party, e.g. Microsoft HealthVault, it is not straightforward that this PHR system will start supporting the identity provider DigiD. This limitation can be resolved as the Dutch government is currently working on a new system that will replace DigiD, called *eID*. This system allows multiple (governmental and commercial) parties to become an official, trusted identity provider that can be used for matters like LSP (Rijksoverheid, 2015).

Interoperability standard The HL7v3 standard can provide a valuable contribution as it fully based on an information model (RIM). Eggebraaten et al. (2007) present in their paper a data model that is based on RIM, which supports the idea of using RIM as the information model

that is jointly agreed upon. However due to the complexity, high implementation costs and low adoption of HL7v3, setting HL7v3 may cause that for example small CDOs or small PHR system developers are unable to join this infrastructure. On the other hand the HL7v2 standard lacks of formality, which will complicate a consistent implementation in an integration scenario where multiple, external parties are cooperating. Furthermore HL7v2 can be problematic regarding global identifiers in order to uniquely identify every piece of information, as explained by Health Level Seven International (2014e). We choose to rely on the HL7 standard FHIR, as described in section 3.2, as it is an open, relatively easy and formal standard, as the underlying model is an abstract the HL7 RIM. By also including the RESTful API, which is an optional part of FHIR, we take a big step towards standardization of health information exchange. As this API results in the fact that every participant of the integration scenario has the exact same interfaces, the level of standardization within the integration infrastructure can become very high. Another advantage is that we have extended the existing architecture of AORTA. Therefore the proposed integration design supports health information exchange between EMR systems equally to the integration of an EMR system and a PHR system.

In order to clarify the architecture, Figure 5.2 illustrates the process of a CDO requesting health information via LSP. Depending on the health information that the authenticated physician wants to view, a specific FHIR request is sent to LSP. LSP then considers which authorization exist for this physician regarding the exchange of health information based on the consents that the patient, i.e. individual, has provided. Subsequently LSP consults the reference index in order to check which connected, external systems contain health information based on the concerned individual, the requested information and the authorizations. The FHIR is then sent to these systems and the responses are received. Concluding the responses are passed back to the requesting CDO and the audit log is updated.

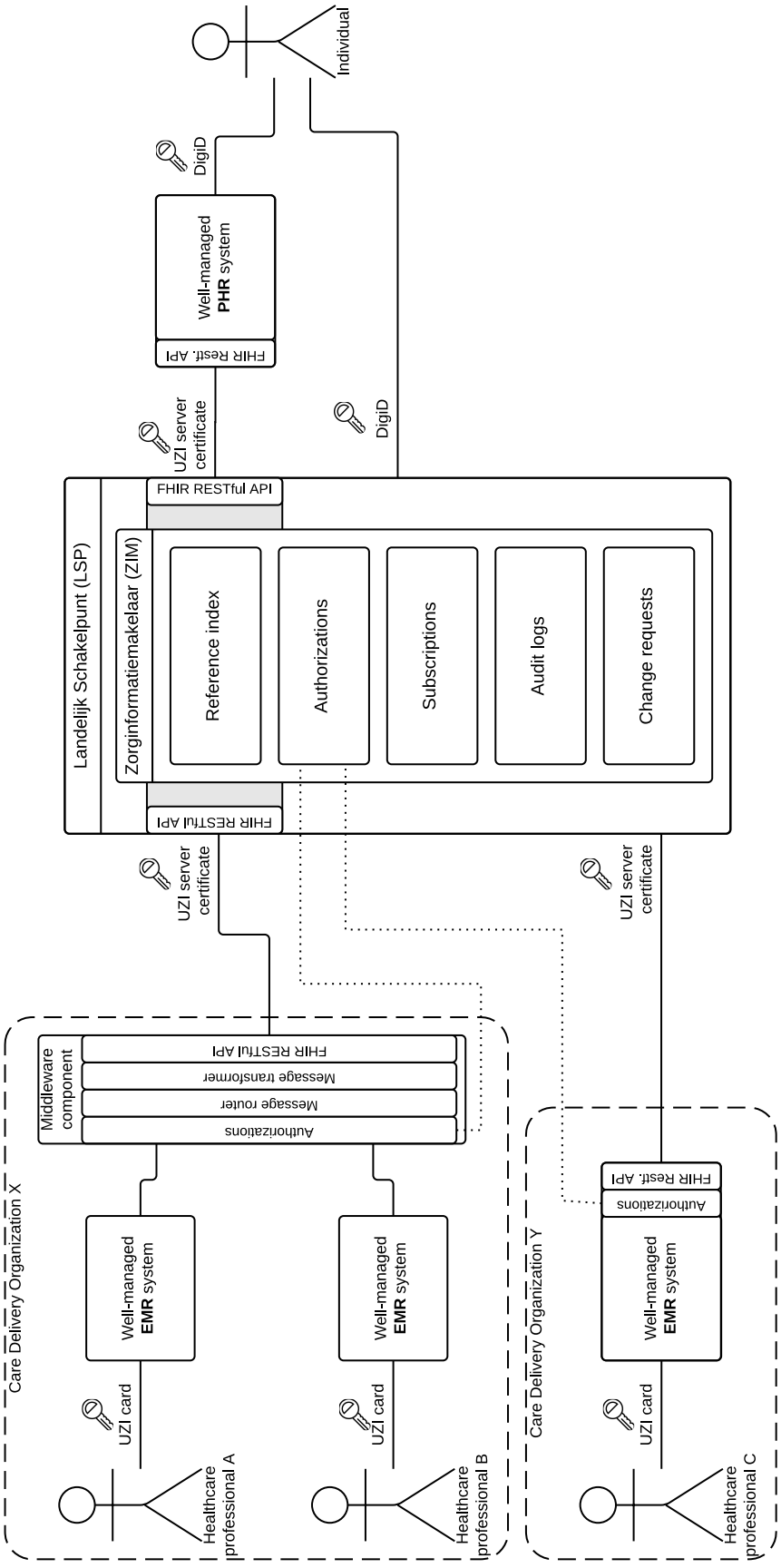


Figure 5.1: Integration architecture overview

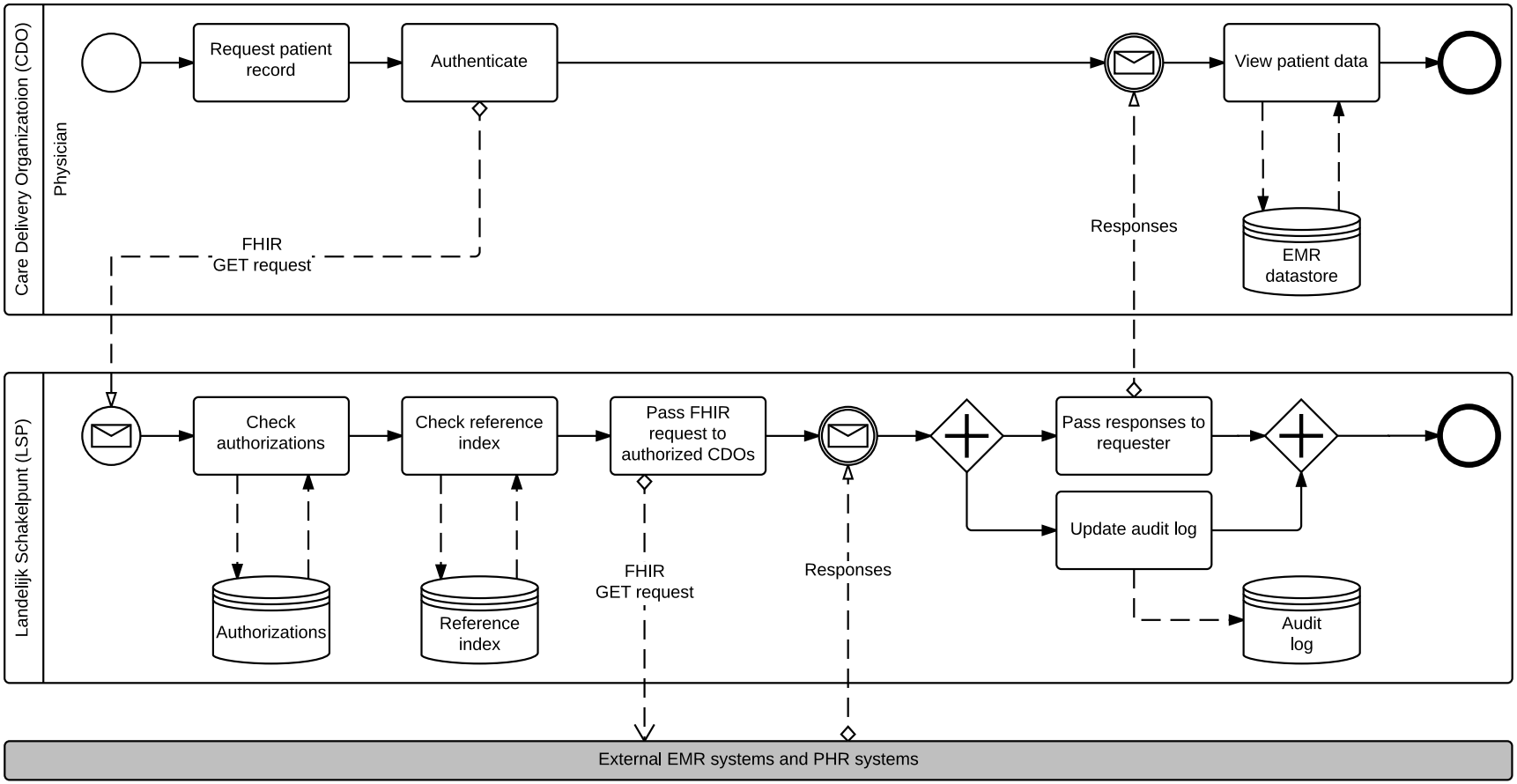


Figure 5.2: Process model for a CDO requesting patient data via LSP

5.4 Roadmap

Due to the fact that FHIR is not released yet, but only a DSTU is available, the first step would be to stimulate and support the development of FHIR, done by HL7. In parallel discussions should be initiated about whether all involved parties are willing to change and expand the current infrastructure, as the success of this integration scenario depends on the full cooperation of every involved party. The concept of FHIR including the RESTful API is a big change compared to the current infrastructure. However having standardized RESTful interfaces will provide lots of different opportunities. An example would be that typically multiple information systems are used within a CDO, e.g. a lab information system or an information system tailored for some specific medical specialty. These information systems often have interfaces with the central EMR system. FHIR could also fulfill a standardizing role for these internal interfaces.
Next to the FHIR support that is required on the side of EMR systems, in parallel VZVZ, the party that maintains LSP, will have to expand the functionality of LSP as described, such that there is a refined consent mechanism in place that can be consulted by every participant for checking authorizations. Furthermore a mechanism should be in place in LSP for supporting *change requests*, such that multiple parties are able to modify health information in a consistent way. The last, but complex, step would be to add support to the EMR and PHR systems for handling data that is received over a virtually integrated system the same way as it would handle locally stored data, but this implementation is up to the developers of EMR and PHR systems.

Chapter 6

Conclusions

The aim of this research is to deliver an architecture for an integrated PHR based on findings derived from theoretical analyses and lessons learned from practice. For the conclusion of this research we go back to the research questions that are defined in section 1.4. The findings, analyses and designs that have been collected and composed in this research enable us to answer these questions.

Which PHR systems are currently available and what are their characteristics? Based on specifications and experiences while using, 68 PHR systems are examined in order to understand their characteristics and the features that they bring. All systems have a cloud, where a user can enter and view his PHR by logging in to an online system. The majority (58 systems) is free-standing. An essential aspect for this research is the presence of an API such that it offers integration possibilities with external systems, e.g. an EMR. Only fifteen systems describes to be equipped with an API, while eight of them are public. A significant part of the analyzed systems offer their service as an *emergency access* mechanism to someone's health record. Often these systems are not intended for the administration of lifelong health records and have limited or no clinical data support. Only two systems, those of Microsoft and Google, are positioned as health information *platforms*, which provide tools to developers to create applications that interact with the platform. Only one system included in the analysis could be classified as an integrated PHR, which shows the current lack of integrated initiatives.

How is health-related data currently shared among different parties? The amount of health-related data that is currently exchanged across multiple CDOs is very limited and is even more restricted when considering exchange in an electronic fashion. Numerous reasons can be found that are currently withhold a proper health information exchange among various CDOs. This is mainly due to the complexity that is inherited by default for such a topic. The exchange of health information is subject to various complicating requirements, such as privacy regulations, interoperability agreements and ownership discussions. Within The Netherlands a health information exchange infrastructure is in place, called AORTA, which is set up as a transport facility. This implies that AORTA does not holds any health information itself, but manages the exchange of data between the systems of exchanging CDO. This exchange has to be authorized by the concerned patient in advance by giving a consent to a CDO to make his EMR available at AORTA. Currently AORTA is primarily used by general practitioners and pharmacists for the exchange of a patient's summarized health record and an overview of current used medication.

What are the lessons learned when prototyping this integrated PHR? In the prototype setup of an integrated PHR, the synchronization of weight measurements was implemented between the PHR system Microsoft HealthVault and the EMR system i.s.h.med. The key lessons learned from this prototype integration are:

- Ensure consent awareness at the source system.

  This implies that any system that contains health information should be able to determine which data has to be exchanged.

- Use an existing interoperability standard.

  The use of a standard is inevitable when exchanging information with external parties. When choosing an existing standard, the scalability of the integration scenario is expected to be bigger as potentially more parties are able to comply with the standard.

- Establish event-driven messaging.

  In order to ensure data integrity and to reduce overhead, source systems preferably present new or updated information actively to a target system, i.e. an event-driven approach.

- Secure scenario integrity.

  Due to the complexity, the sensitivity and required integrity in health information exchange, an integration scenario can only have a minimal, manageable risk for errors by design.

Which challenges arise when we integrate a PHR with an EMR? As described, any form of health information exchange is facing various barriers and difficulties that have to be overcome in order to realize the actual exchange. In the scenario where a PHR is integrated with an EMR, the key challenges can categorized into the following categories:

- Authorization

  The individual should be in control of who can access his PHR. Therefore he should be able to give a consent stating which participant is able to access which part of his PHR. Furthermore the individual has to be able to review audit logs in order to verify which access has been provided to his PHR.

- Authentication

  Due to the sensitivity and the importance of correctness of health information, there should be no doubt regarding the authenticity of participants. Therefore the identity of every participant has to unambiguously verified using a reliable and strong authentication method.

- Interoperability

  In order to ensure data integrity and scalability of integration scenarios, attention should be paid to interoperability setups that reckons with healthcare domain standards, extensible scenarios and best practices within the boundaries of laws and legislation as for example privacy is a direct related topic.

What is the preferred IT architecture in order to realize an integrated PHR? Based on the findings in this research an IT architecture is preferred that relies on virtual links between a central, neutral node and EMR and PHR systems. In this setup EMR systems and PHR systems are not directly linked, but exchange data with the help of this central transport facility. As this architecture shares a lot of similarities with the Dutch AORTA infrastructure, the preferred IT architecture is an extended version of AORTA, containing the following modules:

- Reference index

  Every system indicates which information it has available for exchange of an individual, on the condition that there is a consent allowing this, which is stored in the reference index.

- Authorizations

  Consents of individuals are centrally stored such that systems can consult these to check which data can be or should be exchanged. These consents can be defined on a detailed level, e.g. per data type or per physician.

- Subscriptions

  Systems can subscribe to an individual such that these systems actively receive a notification when some event occurs, e.g. new available data.

- Audit logs

  An individual can view who has access to his health record in order to check whether no access has been granted against his consent.

- Change requests

  Only the creator of a piece of information is allowed to change or delete it. In case another participant wants to have some information changed of which he is not the creator, he may send a change request to the creator.

A requirement for this setup is that it is necessary to have jointly agreed upon a messaging standard, such that information can be exchanged reliably and uniformly. As indicated, the HL7 FHIR standard, including its RESTful API is an appropriate approach for this. Furthermore an authentication method has to be in place for the PHR systems and the individuals. For this respectively UZI server certificates and DigiD can be used. The biggest changes compared to the current version of AORTA is the inclusion of PHR systems in the set of connected systems, the extended *authorizations* module and the principle of change requests.

Which steps should be taken in order to realize this architecture? The first and most important step is to come to an agreement among all involved parties to support the integration of health records. Upon this agreement, a full implementation of HL7 FHIR standards has to be implemented by every participant. Furthermore the functionality of LSP has to be extended with the extended modules *reference index* and *authorizations*, the handling of FHIR communication and the support for connecting PHR systems and individuals. Subsequently, EMR system developers and PHR system developers have to come with a mechanism that merges the data that is retrieved over virtual links with the data from an own data store, which is uniformly presented to the end user. The final step is to include the support for change requests, which requires the system to be able to send these requests and requires LSP to be able to manage these.

When answering the main research question,

How can we link an Electronic Medical Record system with an existing free-standing Personal Health Record system in order to set up an integrated PHR system?

it can be concluded that the integration of PHR systems with EMR systems is a non-trivial, but realistic scenario. However, it requires the full cooperation of every involved party regarding decision-making, patient-centeredness, technical support etcetera. With a virtual integration, consent-based approach, concerns regarding patient privacy, data integrity and health record ownership are properly addressed and ensures the success of integrated PHRs.

6.1 Future research

In this section we discuss the topics or concerns related to this research that require additional research in order to proceed towards an integrated PHR. In the integration design in this research (chapter 5) various new concepts or technological setups are introduced. However, this design does not describe how this implementation is realized on a low, detailed level. For such a detailed implementation directives, additional research is required.

HL7 FHIR As illustrated in Figure 5.1, the architecture greatly relies on the relatively new FHIR standard. FHIR is designed to overcome the adoption barriers of HL7v3, but as FHIR is only released as DSTU and not all resources are currently covered, limited knowledge and experience reports exist about the actual use of FHIR. Additional research and scenario tests will be required in order to support HL7 to make FHIR a solid, mature interoperability standard.

Central consent system In the integration design we discussed the authorizations that an individual can manage centrally in order to control the exchange of his health information. Currently the type of consent that an individual can register within AORTA is binary; Either a CDO is allowed to share an individuals' health information or not. In the integration design a more refined authorization model is required, e.g. which information from a PHR is allowed to be exchanged. However, ideally an individual can manage every form of consent on a single place, which implies that also health information exchange between CDOs can be managed. Such a centralized consent registration concept requires a model that is jointly agreed upon. However setting up a uniform system for the administration of consents is not a straightforward activity. Although a lot of literature exists on this topic (Coiera & Clarke, 2004; Galpottage, 2005; Liu et al., 2013; Powell et al., 2006; Safran et al., 2007; Sicuranza, 2013), only little effort is spend on converging to a standardized, central approach, for example by embedding the HL7 RIM combined with the HL7v3 message for *privacy consent directives* based on CDA (Health Level Seven International, 2011). Within HL7 FHIR the standard for patient consents is currently in development under the *HL7 Patient Friendly Consent Directive Project* and is released as DSTU in January 2015 (Health Level Seven International, 2015).

Requirements engineering Based on the deliverables of this research, a next step would be to start with *requirements engineering*. Based on the directions and guidance that this research offers, the actual detailed design should be supported by an elaborate set of requirements. A good approach for this activity would be to use the *Volere Requirements Specification Template* by Atlantic Systems Guild Ltd. (2004), which gives a structured methodology for requirements engineering activities. The design principles that have been presented in section 5.2 can be used as input for this methodology.

Other aspects In this research we have a strong focus on the technical aspects of the integration design. However, when considering other aspects, various questions may arise:

- To what extent is the integration architecture within the boundaries of local or international laws and legislation, e.g. privacy?
- How do we ensure that CDOs are willing to be part of an integrated health information exchange infrastructure?
- How do we ensure that sufficient resources are available for the design and realization of the infrastructure? "Who is paying?"

- What are the efforts for setting up an international integration scenario and how would it affect the integration design, e.g. in terms of authentication?

In order to ensure a successful integration design, all relevant aspects related to this require consideration.

Recommendations

As a final statement, the recommendation for SAP NL and Cerner would be to start with designing the setup of interoperability interfaces for SAP PAB and i.s.h.med. As the need for healthcare interoperability scenarios is inevitable, SAP NL and Cerner could take a lead in the development of the concept of integrated healthcare. Specifically a leading role is still available for the design of an advanced consent system in which an individual is able to provide his consents in the way he wants in order to put the individual in control. Especially because SAP NL is working on a *consent module* for PAB, opportunities are available to take the lead in the development of a central consent system. Furthermore the potential value of HL7 FHIR together with the lack of success of HL7v3, should trigger SAP NL to start considering the creation of (a part of) the FHIR RESTful API for PAB and i.s.h.med.

Bibliography

Ammenwerth, E., & Winter, A. (2004). *Strategic information management in hospitals: an introduction to hospital information systems*. Springer.

Archer, N., Fevrier-Thomas, U., Lokker, C., McKibbon, K. a., & Straus, S. E. (2011). Personal health records: a scoping review. *Journal of the American Medical Informatics Association : JAMIA*, *18*(4), 515–522. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3128401&tool=pmcentrez&rendertype=abstract doi: 10.1136/amiajnl-2011-000105

Atlantic Systems Guild Ltd. (2004). *Volere Requirements Specification Template - Edition 10.1.*

Bender, D., & Sartipi, K. (2013). HL7 FHIR: An agile and RESTful approach to healthcare information exchange. *Proceedings of CBMS 2013 - 26th IEEE International Symposium on Computer-Based Medical Systems*, 326–331. doi: 10.1109/CBMS.2013.6627810

Benson, T. (2010). *Principles of Interoperability HL7 and Snomed*. Springer.

Bergmann, J., Bott, O. J., Pretschner, D. P., & Haux, R. (2006). An e-consent-based shared EHR system architecture for integrated healthcare networks. *International journal of medical informatics*, *76*(2-3), 130–6. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/16971171 doi: 10.1016/j.ijmedinf.2006.07.013

Bonthuis, M. J. (2007). Privacy en het landelijk Elektronisch Patiënten Dossier (EPD). *Universiteit Groningen*. Retrieved from http://www.itsprivacy.nl/wp-content/uploads/2010/09/privacy-en-het-epd-itp-2008.pdf

Buntin, M. B., Burke, M. F., Hoaglin, M. C., & Blumenthal, D. (2011, March). The benefits of health information technology: a review of the recent literature shows predominantly positive results. *Health affairs (Project Hope)*, *30*(3), 464–71. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/21383365 doi: 10.1377/hlthaff.2011.0178

Calvillo-Arbizu, J. (2014). Standardized access control mechanisms for protecting ISO 13606-based electronic health record systems. *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, 539–542. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6864421

Carrión Señor, I., Fernández-Alemán, J. L., & Toval, A. (2012). Are Personal Health Records Safe ? A Review of Free Web-Accessible Personal Health Record Privacy Policies. *J Med Internet Res*, *14*(4). doi: 10.2196/jmir.1904

Chan, L. (2009). Adoption & Evaluation of Personal Health Record (PHR) System Importance of Healthcare IT. In *Portland international conference on management of engineering & technology, 2009. picmet 2009.* (pp. 2986–2999).

Chan, M., Estève, D., Fourniols, J.-Y., Escriba, C., & Campo, E. (2012, November). Smart wearable systems: current status and future challenges. *Artificial intelligence in medicine*, *56*(3), 137–56. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/23122689 doi: 10.1016/j.artmed.2012.09.003

Coiera, E., & Clarke, R. (2004). e-Consent: The Design and Implementation of Consumer Consent Mechanisms in an Electronic Environment. , *11*(2), 129–141. doi: 10.1197/jamia.M1480.Affiliations

Daglish, D., & Archer, N. (2009, August). Electronic Personal Health Record Systems: A Brief Review of Privacy, Security, and Architectural Issues. *2009 World Congress on Privacy, Security, Trust and the Management of e-Business*, 110–120. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5341709 doi: 10.1109/CONGRESS.2009.14

Detmer, D., Bloomrosen, M., Raymond, B., & Tang, P. (2008, January). Integrated personal health records: transformative tools for consumer-centric care. *BMC medical informatics and decision making*, *8*, 45. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2596104&tool=pmcentrez&rendertype=abstract doi: 10.1186/1472-6947-8-45

Dorr, D., Bonner, L. M., Cohen, A. N., Shoai, R. S., Perrin, R., Chaney, E., & Young, A. S. (2007). Informatics Systems to Promote Improved Care for Chronic Illness: A Literature Review. *Journal of the American Medical Informatics Association*, *14*(2), 156–163. doi: 10.1197/jamia.M2255

Edwards, D. (2014). *nHapi - Home.* Retrieved 2014-11-11, from https://nhapi.codeplex.com/

Eggebraaten, T. J., Tenner, J. W., & Dubbels, J. C. (2007). A health-care data model based on the HL7 Reference Information Model. *IBM Systems Journal*, *46*(1), 5–18. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386594 doi: 10.1147/sj.461.0005

Galpottage, P. a. B. (2005). Patient consent principles and guidelines for e-consent: a New Zealand perspective. *Health Informatics Journal*, *11*, 5–18. doi: 10.1177/1460458205050681

Garets, D., & Davis, M. (2006). Electronic medical records vs. electronic health records: yes, there is a difference. *Policy white paper. Chicago, HIMSS Analytics*, 1–14.

Gillespie, R., Florin, D., & Gillam, S. (2004). How is patient-centred care understood by the clinical, managerial and lay stakeholders responsible for promoting this agenda? *Health expectations : an international journal of public participation in health care and health policy*, *7*(2), 142–148. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/15117388 doi: 10.1111/j.1369-7625.2004.00264.x

Häyrinen, K., Saranto, K., & Nykänen, P. (2008, May). Definition, structure, content, use and impacts of electronic health records: A review of the research literature. *International Journal of Medical Informatics*, *77*(5), 291–304. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17951106 doi: 10.1016/j.ijmedinf.2007.09.001

Health Level Seven International. (2005). *HL7 Clinical Document Architecture, Release 2.0* (Tech. Rep.). Retrieved from http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7

Health Level Seven International. (2009). *Mappings and Translations - HL7Wiki.* Retrieved 2015-03-06, from http://wiki.hl7.org/index.php?title=Mappings_and_Translations

Health Level Seven International. (2011). *HL7 Implementation Guide for Clinical Document Architecture , Release 2 : Consent Directives , Release 1* (No. January).

Health Level Seven International. (2012). *HL7 Reference Information Model v2.41.* Retrieved from http://www.hl7.org/implement/standards/rim.cfm

Health Level Seven International. (2013). *HL7 Version 3 Detailed Clinical Models, Release 1* (No. January).

Health Level Seven International. (2014a). *About Health Level Seven.* Retrieved from http://www.hl7.org/about/index.cfm?ref=nav

Health Level Seven International. (2014b). *HL7 Personal Health Record System Functional Model* (Tech. Rep. No. May).

Health Level Seven International. (2014c). *HL7 Version 3 Product Suite.* Retrieved 2015-02-11, from http://www.hl7.org/implement/standards/product_brief.cfm?product_id=186

Health Level Seven International. (2014d). *The Relationship between FHIR and other HL7 Standards - FHIR v.0.0.82.* Retrieved 2015-03-05, from http://www.hl7.org/implement/standards/fhir/comparison.html

Health Level Seven International. (2014e). *Relationship between FHIR and v2 Messaging - FHIR v0.0.82.* Retrieved 2015-02-28, from http://www.hl7.org/implement/standards/fhir/comparison-v2.html

Health Level Seven International. (2014f). *Relationship between FHIR and v3 Messaging - FHIR v0.0.82.* Retrieved 2015-02-28, from http://www.hl7.org/implement/standards/fhir/comparison-v3.html

Health Level Seven International. (2015). *HL7 Patient Friendly Consent Directive Project.* Retrieved 2015-03-04, from http://wiki.hl7.org/index.php?title=HL7_Patient_Friendly_Consent_Directive_Project

Hinman, A. R., & Ross, D. A. (2010). Immunization Registries Can Be Building Blocks For National Health Information Systems. *Health Affairs*, *29*(4), 676–682. Retrieved from http://www.commed.vcu.edu/IntroPH/Communicable_Disease/2010/registryasbase4pract.pdf

Hodes, W. (2013). Gedragscode Elektronische Gegevensuitwisseling in de Zorg.

Hohpe, G., & Woolf, B. (2004). *Enterprise integration patterns: Designing, building, and deploying messaging solutions.* Addison-Wesley Professional.

IBM. (2004). *UML Basics: The sequence diagram.* Retrieved from http://www.ibm.com/developerworks/rational/library/3101.html

Kaelber, D. C., Jha, A. K., Johnston, D., Middleton, B., & Bates, D. W. (2008). A research agenda for personal health records (PHRs). *Journal of the American Medical Informatics Association : JAMIA*, *15*(6), 729–736. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2585530&tool=pmcentrez&rendertype=abstract doi: 10.1197/jamia.M2547.Introduction

Khan, A. (2013). Wearable Health Monitoring System. *HCI International 2013-Posters' Extended Abstracts*, 173–177. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-39476-8_36

Klaus, H., Rosemann, M., & Gable, G. G. (2000). What is ERP? *Information systems frontiers*, *2*(2), 141–162.

Koeken, M. (2012). *Identifying and ranking the selection criteria in European tenders regarding the procurement of HIS / EHR systems in Dutch academic hospitals* (Tech. Rep.). Eindhoven: Eindhoven University of Technology.

Little, P., Everitt, H., Williamson, I., Warner, G., Moore, M., Gould, C., ... Payne, S. (2001). Observational study of effect of patient centredness and positive approach on outcomes of general practice consultations. *BMJ (Clinical research ed.)*, *323*, 908–911. doi: 10.1136/bmj.323.7318.908

Liu, C.-H., Lin, F.-Q., Chiang, D.-L., Chen, T.-S. T.-L., Chen, C.-S., Lin, H.-Y., & Chung, Y.-F. (2013, October). Secure PHR Access Control Scheme for Healthcare Application Clouds. *2013 42nd International Conference on Parallel Processing*, 1067–1076. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6687452 doi: 10.1109/ICPP.2013.127

Madra, J., Pool, K., Roberts, J., & Savage, R. (2013). *HL7 Version 2.5.1 Implementation Guide : Height and Weight Report , Release 1 - US Realm* (Tech. Rep.). Health Level Seven International.

Microsoft. (2014a). *Data types - HealthVault Developer Tools.* Retrieved 2015-02-07, from https://developer.healthvault.com/DataTypes

Microsoft. (2014b). *HealthVault - Apps & devices.* Retrieved 2014-09-11, from https://account.healthvault.com/us/en/Directory?target=Devices

Microsoft. (2014c). *Methods - HealthVault Developer Tools.* Retrieved 2015-02-10, from https://developer.healthvault.com/Methods

Microsoft. (2015a). *Establishing connectivity with HealthVault users.* Retrieved 2015-02-16, from https://msdn.microsoft.com/en-us/healthvault/dn800879

Microsoft. (2015b). *HealthVault Eventing.* Retrieved 2015-02-20, from https://msdn.microsoft.com/en-us/healthvault/dn799024

Ministerie van VWS. (2012). Toelichting doorstart LSP. (april 2012), 1–6.

Mukherjee, S., Dolui, K., & Datta, S. (2014). Patient health management system using e-health monitoring architecture. *Advance Computing Conference (IACC), 2014 IEEE International*, 400–405. Retrieved from http://www.google.com/patents/US5729472http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6779357

Nederlandse Patiënten Consumenten Federatie. (2014). Notitie Overzicht Nederlandse PGD-platforms. (November), 1–8.

Nederlandse Zorgautoriteit. (2014). *Update DBC-pakket 2015 (RZ15b), stollingsfactoren en ICD-10* (Tech. Rep.). Retrieved from http://www.nza.nl/1048076/1048107/CI_14_62c__Update_DBC_pakket_2015_RZ15b_stollingsfactoren_en_ICD_10.pdf

Nictiz. (2012a). *Alles wat je wilt (moet) weten over CCR/CCD* (Tech. Rep. No. september). Author.

Nictiz. (2012b). ICT-standaarden in de zorg.

Nictiz. (2012c). *Professionele samenvatting via de infrastructuur tijdens dienstwaarneming.*

Nictiz. (2014a). *Corporate brochure.*

Nictiz. (2014b). *Over Nictiz.* Retrieved from http://www.nictiz.nl/page/Over-Nictiz/English

Nike. (2014). *Nike Digital Privacy Policy.* Retrieved from http://www.nike.com/us/en_us/c/help/privacy-policy

Pluut, B. (2010). *Het landelijk EPD als blackbox* (Vol. 45) (No. 45).

Powell, J., Fitton, R., & Fitton, C. (2006). Sharing electronic health records: The patient view. *Informatics in Primary Care*, *14*, 55–57.

Puustjarvi, J., & Puustjarvi, L. (2010). Providing relevant health information to patient-centered healthcare. In *e-health networking applications and services (healthcom), 2010 12th ieee international conference on* (pp. 215–220). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5556569

Raad voor de Volksgezondheid en Zorg. (2014). *Patiënteninformatie RVZ raad in gezondheidszorg.* Den Haag.

Rechtbank Midden-Nederland. (2014). *ECLI : NL : RBMNE : 2014 : 3097.* Retrieved from http://uitspraken.rechtspraak.nl/inziendocument?id=ECLI:NL:RBMNE:2014:3097

Reichertz, P. (2006). Hospital information systems—Past, present, future. *International Journal of Medical Informatics*, *75*(3), 282–299. Retrieved from http://www.sciencedirect.com/science/article/pii/S1386505605002091 doi: 10.1016/j.ijmedinf.2005.10.001

Rijksoverheid. (2015). *eID Stelsel.* Retrieved 2015-03-03, from http://www.eid-stelsel.nl/

Rø stad, L. (2008, March). An initial model and a discussion of access control in patient controlled health records. In *Ares 2008 - 3rd international conference on availability, security, and reliability, proceedings* (pp. 935–942). Ieee. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4529444 doi: 10.1109/ARES.2008.185

Safran, C., Bloomrosen, M., Hammond, W., Labkoff, S., Markel-Fox, S., Tang, P., ... Others (2007). Toward a national framework for the secondary use of health data: an American Medical Informatics Association white paper. *Journal of the American Medical Informatics Association*, *14*(1), 1. Retrieved from http://jamia.bmj.com/content/14/1/1.short doi: 10.1197/jamia.M2273.Introduction

Sahay, B., & Gupta, A. (2003). Development of software selection criteria for supply chain solutions. *Industrial Management & Data Systems*, *103*(2), 97–110. Retrieved from http://www.emeraldinsight.com/10.1108/02635570310463429 doi: 10.1108/02635570310463429

SAP. (2014). *Supported Databases.* Retrieved from http://help.sap.com/saphelp_nw73/helpdata/en/84/0cb892edfbc34290c1685132006662/frameset.htm

SAP SE. (2013). *SAP HANA Platform for Healthcare: Bringing the World Closer to Real-Time Personalized Medicine.* Retrieved 2014-03-03, from https://blogs.saphana.com/2013/10/15/sap-hana-for-healthcare-bringing-the-world-closer-to-real-time-personalized-medicine/

SAP SE. (2014a). *Components of SAP Communication Technology.* Retrieved 2015-02-17, from https://help.sap.com/saphelp_nw70/helpdata/en/7c/a4f1b3c59aef4f8ea3c32cda0c0486/content.htm

SAP SE. (2014b). *Healthcare — Value Map — SAP solution explorer.* Retrieved from https://rapid.sap.com/se/executive#!/vlm/i_healthcr/vlm-i_healthcr-ind~~.~~i_healthcr

Schuldt, H. (2009). Multi-Tier Architecture. In *Encyclopedia of database systems* (pp. 1862–1865). Springer.

Sicuranza, M. (2013). An Access Control Model for easy management of patient privacy in ERR systems. , 463–470.

Siemens AG. (2014). *i.s.h.med basis.* Retrieved from http://www.healthcare.siemens.com/hospital-it/clinical-information-systems/clinical-information-systems-ishmed/ishmed-basis

Sonoda, T. (2011). Evolution of Electronic Medical Record Solutions. *Fujitsu Sci. Tech. J*, *47*(1), 19–27.

Tang, P. C., Ash, J. S., Bates, D. W., Overhage, J. M., & Sands, D. Z. (2006). Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. *Journal of the American Medical Informatics Association*, *13*(2), 121–126. Retrieved from http://jamia.bmj.com/content/13/2/121.short doi: 10.1197/jamia.M2025.records

The National Alliance for Health Information Technology. (2008). *Defining Key Health Information Technology Terms* (Tech. Rep.).

Van Gorp, P., & Comuzzi, M. (2012). MyPHRMachines: Lifelong personal health records in the cloud. In *Proceedings - ieee symposium on computer-based medical systems* (pp. 1–6). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6266378 doi: 10.1109/CBMS.2012.6266378

van Strien, P. J. (1997). Towards a methodology of psychological practice the regulative cycle. *Theory & Psychology*, *7*(5), 683–700.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A Break in the Clouds : Towards a Cloud Definition. *ACM SIGCOMM Computer Communication Review*, *39*(1), 50–55.

VZVZ. (2013). *Architectuur AORTA* (Tech. Rep.).

Wagner, M., & Jones, J. (2009). Using CCR and CCD with HealthVault. In *Microsoft connected health conference.*

Wisconsin State Journal. (2015). *Epic Systems to open its own app exchange.* Retrieved 2015-02-25, from http://host.madison.com/business/epic-systems-to-open-its-own-app-exchange/article_fc7e8b94-b1ec-59f4-9065-1e6143fe351c.html

World Health Organization. (2015). *ICD-10 Version:2015.* Retrieved from http://apps.who.int/classifications/icd10/browse/2015/en

Zorgvisie. (2014). *Overzicht epd/ZIS in ziekenhuizen.* Retrieved from http://www.zorgvisie.nl/ICT/Overzicht-epdZIS-in-ziekenhuizen/?cmpid=NLC|zorgvisie_ict|Epd/ZIS-leveranciers_per_ziekenhuis

Appendix A

Types of PHR system architectures

In the figures A.1, A.2, A.3 and A.4 a simplified model of every PHR system architecture are shown.

Figure A.1: Freestanding PHR system

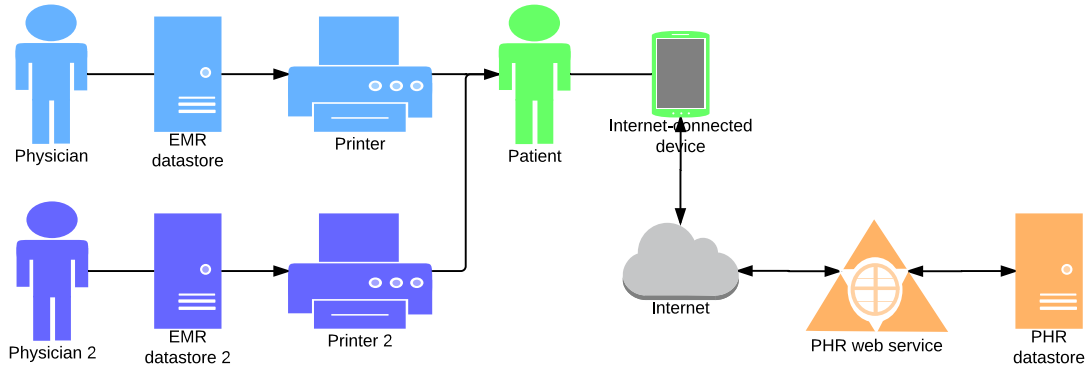


Figure A.2: Provider-tethered PHR system

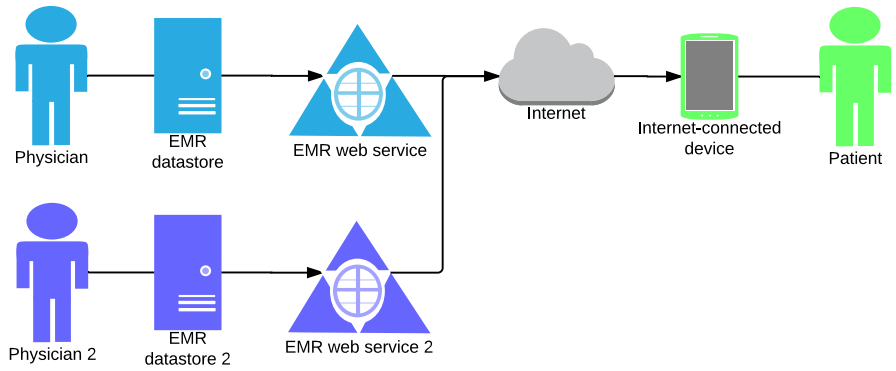


Figure A.3: Claims-tethered PHR system

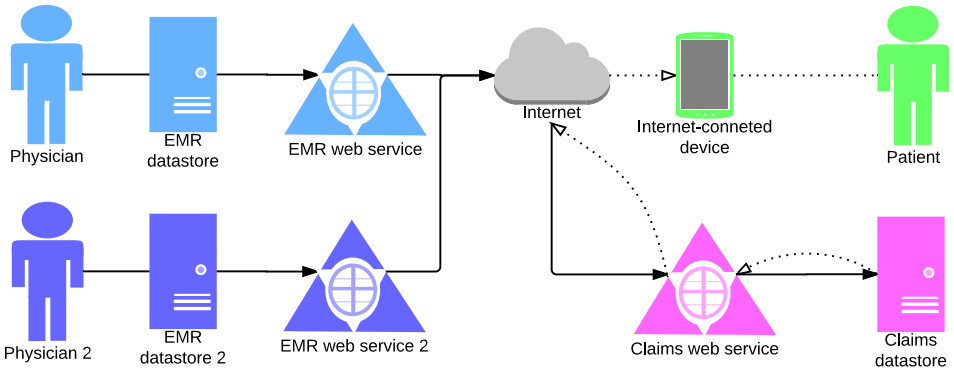
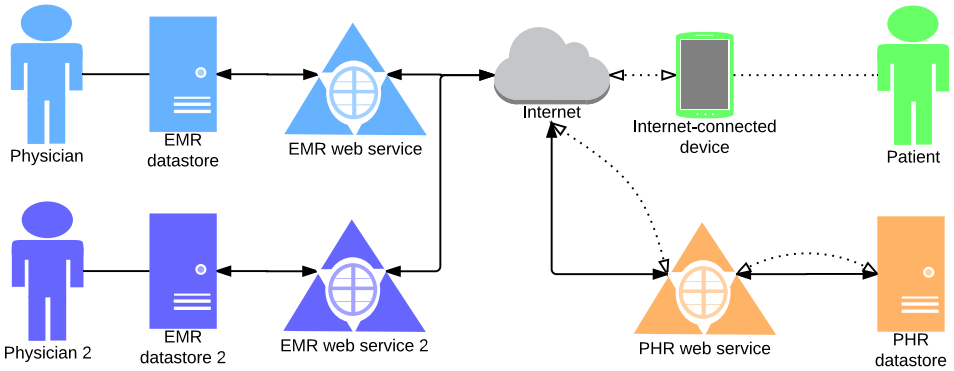


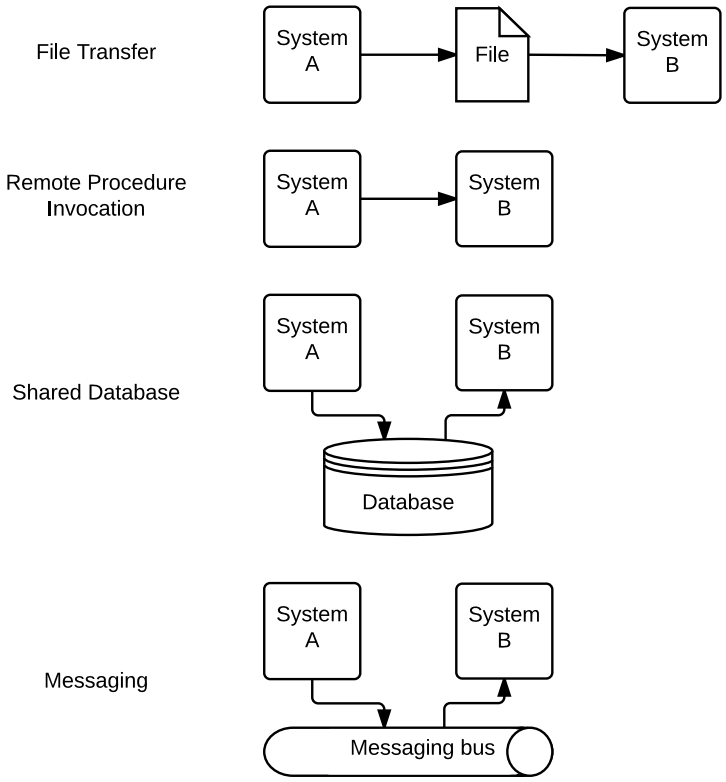
Figure A.4: Integrated PHR system



Appendix B

Interoperability patterns

Figure B.1: Interoperability patterns



Appendix C

PHR analysis results

Effective date of analysis: October 29, 2014.

Table C.1: PHR systems analysis

	Name	API	PHR type	Pricing model	Clinical data support	CCR/CCD import-export support	Microsoft Healthvault integration	Data owner	Dutch language support	Developers community	Status	Mobile device support
1	AccessMyRecords	Public	Free-stand.	Paid (recur.)	Yes	Yes	No	Supplier	No	No	Active	Yes
2	Activ Doctors Online	None	Free-stand.	Paid (recur.)	Yes	No	No	Supplier	No	No	Active	Yes
3	AMESMyFile	None	Free-stand.	Paid (recur.)	Yes	No	No	Customer	No	No	Active	No
4	Capzule PHR	None	Free-stand.	Paid (one-t.)	No	No	Yes	Supplier	No	No	Active	Yes
5	CareSync	None	Free-stand.	Free w/ p.u. (recur. & one-t.)	Yes	No	No	Supplier	No	No	Active	Yes
6	Dossia	Public	Prov.-teth.	Sponsered	Yes	Yes	No	Patient	No	No	Active	Yes
7	Drimpy	None	Free-stand.	Free w/ p.u. (one-t.)	No	No	No	Patient	Yes	No	Active	No
8	Epic MyChart	Private	Prov.-teth.	Sponsered	Yes	Export	No	Customer	No	No	Active	Yes
9	FollowMyHealth	None	Free-stand.	Free	Yes	Yes	No	Supplier	No	No	Active	Yes
10	GlobalPatientRecord	None	Free-stand.	Paid (recur.)	Yes	No	No	Patient	No	No	Active	No
11	Google Fit	Public	Free-stand.	Free	No	No	No	Supplier	No	Yes	Active	Yes
12	Health Companion	None	Free-stand.	Free	Yes	Import	No	Supplier	No	No	Active	Yes
13	HealtheTracks	None	Free-stand.	Paid (recur.)	No	No	No	Patient	No	No	Active	No
14	Healthgram.com	None	Prov.-teth.	Paid (recur.)	Yes	No	No	Customer	No	No	Active	Yes
15	Healthspek	None	Free-stand.	Free	Yes	Import	No	Supplier	No	No	Active	Yes

Table C.1: PHR systems analysis

	Name	API	PHR type	Pricing model	Clinical data support	CCR/CCD import-export support	Microsoft Healthvault integration	Data owner	Dutch language support	Developers community	Status	Mobile device support
16	HealthString	None	Free-stand.	Paid (recur.)	No	No	No	Supplier	No	No	Active	No
17	HealthTrio	None	Free-stand.	Paid (recur.)	Yes	Yes	No	Supplier	No	No	Active	No
18	It Runs in My Family	None	Free-stand.	Free	No	No	No	Unknown	No	No	Active	No
19	Juniper Health	None	Free-stand.	Paid (recur.)	No	No	No	Supplier	No	No	Active	No
20	KIS EHR	None	Free-stand.	Free	Yes	No	No	Unknown	No	No	Active	No
21	LifeLedger	None	Free-stand.	Paid (recur.)	No	No	No	Unknown	No	No	Active	No
22	LifeOnKey	None	Free-stand.	Paid (recur.)	Yes	Yes	No	Supplier	No	No	Active	No
23	Lynxcare	None	Free-stand.	Paid (one-t.)	Yes	No	No	Supplier	No	No	Active	No
24	Magnus Health Portal	None	Free-stand.	Paid (recur.)	Yes	No	No	Supplier	No	No	Active	Yes
25	MedDataNet	None	Free-stand.	Paid (one-t.)	No	No	No	Supplier	No	No	Active	No
26	MedicAlert	None	Free-stand.	Paid (one-t.)	No	No	No	Supplier	No	No	Active	Yes
27	MediKeeper	Public	Free-stand.	Sponsered	Yes	Yes	No	Supplier	No	No	Active	Yes
28	MediKeeper (Healthvault)	Public	Free-stand.	Free	Yes	Yes	Yes	Supplier	No	No	Active	Yes
29	MedischeGegevens.nl	Private	Prov.-teth.	Sponsered	Yes	No	No	Customer	Yes	No	Active	No
30	MedNotice	None	Free-stand.	Free	No	No	No	Supplier	No	No	Active	No
31	MedStick	None	Free-stand.	Paid (one-t.)	No	No	No	Patient	Yes	No	Active	No
32	Microsoft HealthVault	Public	Free-stand.	Free	Yes	Yes	Yes	Supplier	Yes	Yes	Active	Yes

Table C.1: PHR systems analysis

	Name	API	PHR type	Pricing model	Clinical data support	CCR/CCD import-export support	Microsoft Healthvault integration	Data owner	Dutch language support	Developers community	Status	Mobile device support
33	MijnGezondheid.net	None	Prov.-teth.	Free	No	No	No	Unknown	Yes	No	Active	No
34	MijnZorgnet	None	Free-stand.	Free	Yes	No	No	Patient	Yes	No	Active	No
35	MiVIA	None	Prov.-teth.	Sponsered	Yes	Import	No	Patient	No	No	Active	No
36	My Doclopedia PHR	None	Free-stand.	Free	Yes	No	No	Supplier	No	No	Active	No
37	MyAlert	Private	Free-stand.	Paid (recur.)	Yes	No	No	Supplier	Yes	No	Active	Yes
38	MyHealth Online	None	Free-stand.	Sponsered	No	No	No	Customer	Yes	No	Active	No
39	myhealthaccount.com	Public	Free-stand.	Free	Yes	No	No	Unknown	No	No	Active	Yes
40	myHealthFolders	None	Free-stand.	Sponsered	No	No	No	Supplier	No	No	Active	No
41	MyLifeSaver	None	Free-stand.	Paid (recur.)	Yes	No	No	Unknown	No	No	Active	No
42	MyMedicalRecords.com	None	Free-stand.	Paid (recur.)	Yes	No	No	Patient	No	No	Active	No
43	MyMedWall	Private	Free-stand.	Free	Yes	Yes	No	Supplier	No	No	Active	Yes
44	NoMoreClipBoard.com	Private	Free-stand.	Free w/ p.u. (recur.)	Yes	Export	No	Supplier	No	No	Active	Yes
45	OnlineMedicalRegistry	None	Free-stand.	Paid (one-t.)	No	No	No	Patient	No	No	Active	No
46	Patiënt1	None	Free-stand.	Free	Yes	No	No	Patient	Yes	No	Active	No
47	Patient Power	None	Free-stand.	Free w/ p.u. (one-t.)	Yes	No	No	Unknown	No	No	Active	Yes

Table C.1: PHR systems analysis

	Name	API	PHR type	Pricing model	Clinical data support	CCR/CCD import-export support	Microsoft Healthvault integration	Data owner	Dutch language support	Developers community	Status	Mobile device support
48	Patients know best	Public	Free-stand.	Paid (recur.)	Yes	No	No	Patient	No	No	Active	Yes
49	Pazio	Private	Integrated	Sponsered	Yes	No	No	Unknown	Yes	No	Active	No
50	People Chart	None	Prov.-teth.	Sponsered	Yes	No	No	Unknown	No	No	Active	No
51	RelayHealth	Public	Free-stand.	Free	Yes	Yes	Yes	Customer	No	No	Active	No
52	RememberItNow!	None	Free-stand.	Free	No	No	No	Supplier	No	No	Active	Yes
53	Synchart	None	Free-stand.	Paid (recur.)	No	No	No	Unknown	No	No	Active	No
54	WebMD Health Manager	None	Free-stand.	Free	Yes	No	No	Supplier	No	No	Active	No
55	WellCase	None	Free-stand.	Paid (recur.)	No	No	No	Supplier	No	No	Active	No
56	Your Health Record	None	Free-stand.	Free	Yes	No	No	Supplier	No	No	Active	Yes
57	Zorgdoc	None	Free-stand.	Free	No	No	No	Patient	Yes	No	Active	Yes
58	ZweenaHealth	Private	Free-stand.	Paid (recur.)	Yes	Import	Yes	Supplier	No	No	Active	No
59	A Smart PHR										In dev.	
60	Avado										Discont.	
61	CapMedPHR										Discont.	
62	iHealthRecord										Discont.	
63	Medical ID Card										Discont.	
64	MedicalSummary										Discont.	

Appendix D

Health Level 7

Figure D.1: Common HL7v2 message segments (Benson, 2010)

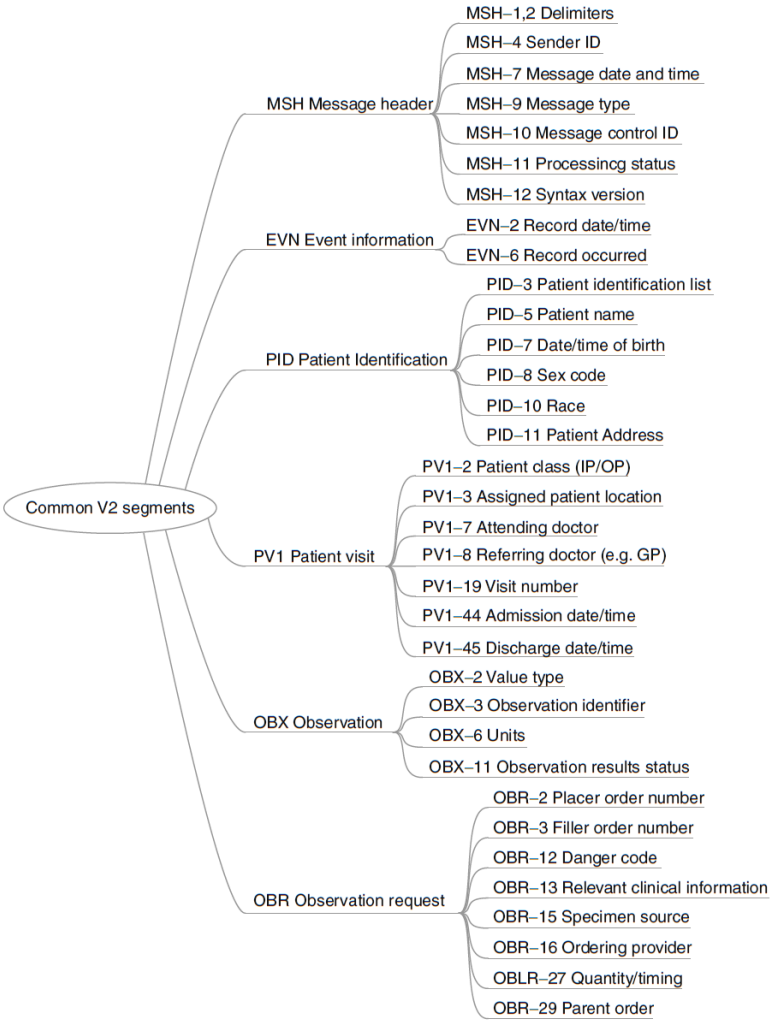
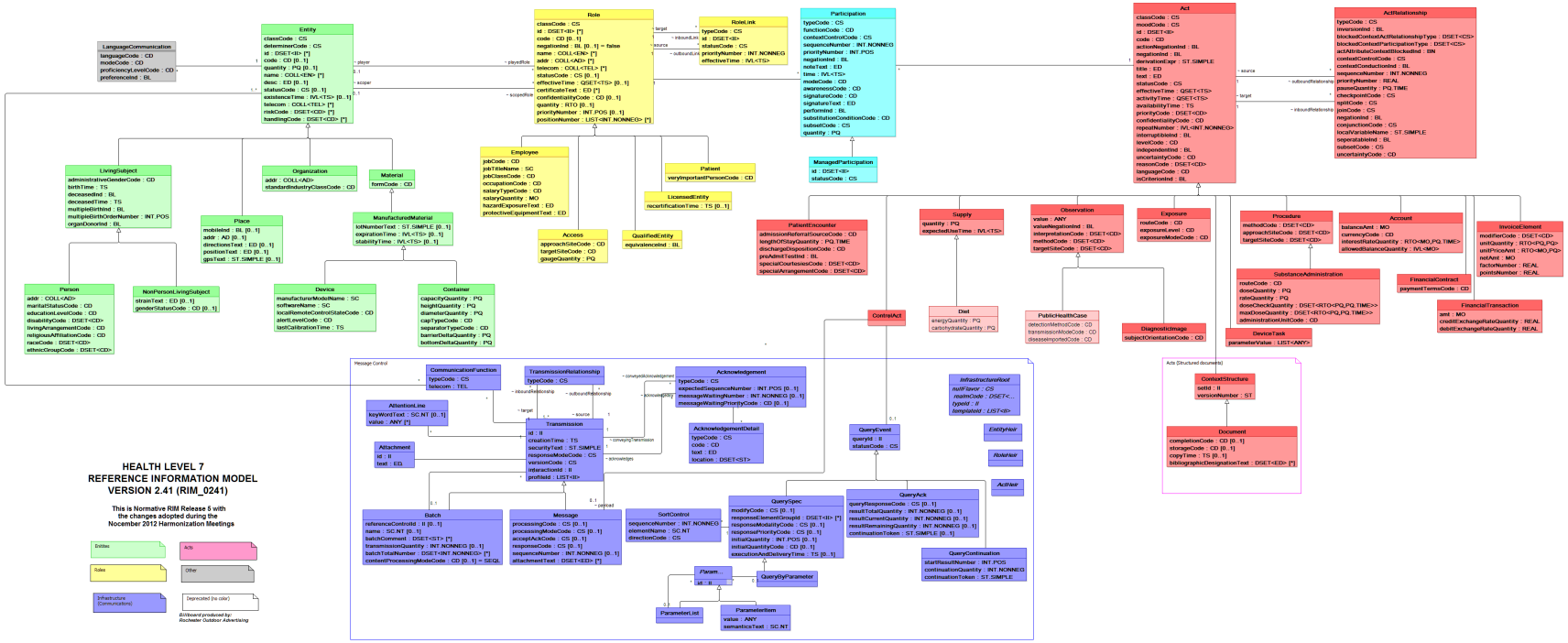


Figure D.2: HL7 Reference Information Model v2.41 (Health Level Seven International, 2012)



Listing D.1: Example HL7v3 message based on CDA

```
<ClinicalDocument>
       <code code="3142-7" displayName="BODY WEIGHT (STATED)" codeSystem="
           ↪ 2.16.840.1.113883.6.1" codeSystemName="LOINC"/>
       <title>Body Weight</title>
       <text>
              <paragraph>
                     <content ID='content-1'>285 pounds</content>
              </paragraph>
       </text>
       <entry>
              <observation classCode='OBS' moodCode='EVN'>
                     <code code="3142-7" displayName="BODY WEIGHT (STATED)"
                         ↪ codeSystem="2.16.840.1.113883.6.1" codeSystemName="
                         ↪ LOINC"/>
                     <effectiveTime value=""/>
                     <text>
                            <reference value="#content-1" />
                     </text>
                     <value xsi:type="PQ" value="285" unit="[lb_us]" />
              </observation>
       </entry>
</ClinicalDocument>
```

Listing D.2: HL7v3 CCD sample, XML (Health Level Seven International, 2009)

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="CCD.xsl"?>
<!-- The following sample document depicts a fictional character's health
    ↪ summary. Any resemblance to a real person is coincidental. -->
<ClinicalDocument xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc" xmlns:
    ↪ xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:
    ↪ hl7-org:v3 CDA.xsd">
       <!--
**********************************************************
CDA Header
**********************************************************
-->
       <typeID root="2.16.840.1.113883.1.3" extension="POCD_HD000040"/>
       <templateId root="2.16.840.1.113883.10.20.1"/> <!-- CCD v1.0 Templates
           ↪ Root -->
       <id root="db734647-fc99-424c-a864-7e3cda82e703"/>
       <code code="34133-9" codeSystem="2.16.840.1.113883.6.1" displayName="
           ↪ Summarization of episode note"/>
       <title>Good Health Clinic Continuity of Care Document</title>
       <effectiveTime value="20000407130000+0500"/>
       <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
       <languageCode code="en-US"/>
       <recordTarget>
```

```
        <patientRole>
            <id extension="996-756-495" root="2.16.840.1.113883.19.5"/
                ↪ >
            <patient>
                <name>
                    <given representation="TXT" mediaType="text
                        ↪ /plain" partType="GIV">Henry</given>
                    <family representation="TXT" mediaType="
                        ↪ text/plain" partType="FAM">Levin</
                        ↪ family>
                    <suffix representation="TXT" mediaType="
                        ↪ text/plain" partType="SUF">the 7th</
                        ↪ suffix>
                </name>
                <administrativeGenderCode code="M" codeSystem="
                    ↪ 2.16.840.1.113883.5.1"/>
                <birthTime value="19320924"/>
            </patient>
            <providerOrganization>
                <id root="2.16.840.1.113883.19.5"/>
                <name>Good Health Clinic</name>
            </providerOrganization>
        </patientRole>
    </recordTarget>
    <author>
        <time value="20000407130000+0500"/>
        <assignedAuthor>
            <id root="20cf14fb-b65c-4c8c-a54d-b0cca834c18c"/>
            <assignedPerson>
                <name><prefix>Dr.</prefix><given>Robert</given><
                    ↪ family>Dolin</family></name>
            </assignedPerson>
            <representedOrganization>
                <id root="2.16.840.1.113883.19.5"/>
                <name>Good Health Clinic</name>
            </representedOrganization>
        </assignedAuthor>
    </author>
    <informant>
        <assignedEntity>
            <id nullFlavor="NI"/>
            <representedOrganization>
                <id root="2.16.840.1.113883.19.5"/>
                <name>Good Health Clinic</name>
            </representedOrganization>
        </assignedEntity>
    </informant>
    <custodian>
        <assignedCustodian>
```

```
                <representedCustodianOrganization>
                        <id root="2.16.840.1.113883.19.5"/>
                        <name>Good Health Clinic</name>
                </representedCustodianOrganization>
        </assignedCustodian>
</custodian>
<legalAuthenticator>
        <time value="20000407130000+0500"/>
        <signatureCode code="S"/>
        <assignedEntity>
                <id nullFlavor="NI"/>
                <representedOrganization>
                        <id root="2.16.840.1.113883.19.5"/>
                        <name>Good Health Clinic</name>
                </representedOrganization>
        </assignedEntity>
</legalAuthenticator>
<participant typeCode="IND">
        <associatedEntity classCode="GUAR">
                <id root="4ff51570-83a9-47b7-91f2-93ba30373141"/>
                <addr>
                        <streetAddressLine>17 Daws Rd.</streetAddressLine>
                        <city>Blue Bell</city>
                        <state>MA</state>
                        <postalCode>02368</postalCode>
                </addr>
                <telecom value="tel:(888)555-1212"/>
                <associatedPerson>
                        <name>
                                <given>Kenneth</given>
                                <family>Ross</family>
                        </name>
                </associatedPerson>
        </associatedEntity>
</participant>
<participant typeCode="IND">
        <associatedEntity classCode="NOK">
                <id root="4ac71514-6a10-4164-9715-f8d96af48e6d"/>
                <code code="65656005" codeSystem="2.16.840.1.113883.6.96"
                    ↪ displayName="Biiological mother"/>
                <telecom value="tel:(999)555-1212"/>
                <associatedPerson>
                        <name>
                                <given>Henrietta</given>
                                <family>Levin</family>
                        </name>
                </associatedPerson>
        </associatedEntity>
</participant>
```

```
        <documentationOf>
                <serviceEvent classCode="PCPR">
                        <effectiveTime><low value="19320924"/><high value="
                            ↪ 20000407"/></effectiveTime>
                        <performer typeCode="PRF">
                                <functionCode code="PCP" codeSystem="
                                    ↪ 2.16.840.1.113883.5.88"/>
                                <time><low value="1990"/><high value='20000407'/><
                                    ↪ /time>
                                <assignedEntity>
                                        <id root="20cf14fb-b65c-4c8c-a54d-
                                            ↪ b0cca834c18c"/>
                                        <assignedPerson>
                                                <name><prefix>Dr.</prefix><given>
                                                    ↪ Robert</given><family>Dolin</
                                                    ↪ family></name>
                                        </assignedPerson>
                                        <representedOrganization>
                                                <id root="2.16.840.1.113883.19.5"/>
                                                <name>Good Health Clinic</name>
                                        </representedOrganization>
                                </assignedEntity>
                        </performer>
                </serviceEvent>
        </documentationOf>
        <!--
******************************************************
CDA Body
******************************************************
-->
        <component>
                <structuredBody>
                        <!--
******************************************************
Vital Signs section
******************************************************
-->
<component>
<section>
        <templateId root="2.16.840.1.113883.10.20.1.16"/> <!-- Vital signs
            ↪ section template -->
        <code code="8716-3" codeSystem="2.16.840.1.113883.6.1"/>
        <title>Vital Signs</title>
        <text>
                <table border="1" width="100%">
                        <thead>
                                <tr><th align="right">Date / Time: </th><th>Nov
                                    ↪ 14, 1999</th><th>April 7, 2000</th></tr>
                        </thead>
```

```
                <tbody>
                        <tr><th align="left">Height</th><td>177 cm</td><td
                            ↪ >177 cm</td></tr>
                        <tr><th align="left">Weight</th><td>86 kg</td><td>
                            ↪ 88 kg</td></tr>
                        <tr><th align="left">Blood Pressure</th><td>132/86
                            ↪  mmHg</td><td>145/88 mmHg</td></tr>
                </tbody>
        </table>
</text>
<entry typeCode="DRIV">
        <organizer classCode="CLUSTER" moodCode="EVN">
                <templateId root="2.16.840.1.113883.10.20.1.35"/> <!--
                    ↪ Vital signs organizer template -->
                <id root="c6f88320-67ad-11db-bd13-0800200c9a66"/>
                <code code="46680005" codeSystem="2.16.840.1.113883.6.96"
                    ↪ displayName="Vital signs"/>
                <statusCode code="completed"/>
                <effectiveTime value="19991114"/>
                <component>
                        <observation classCode="OBS" moodCode="EVN">
                                <templateId root="
                                    ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                                    ↪  Result observation template -->
                                <id root="c6f88321-67ad-11db-bd13-0800200
                                    ↪ c9a66"/>
                                <code code="50373000" codeSystem="
                                    ↪ 2.16.840.1.113883.6.96" displayName=
                                    ↪ "Body height"/>
                                <statusCode code="completed"/>
                                <effectiveTime value="19991114"/>
                                <value codeSystem="2.16.840.1.113883.6.8"
                                    ↪ xsi:type="PQ" value="177" unit="cm"/
                                    ↪ >
                        </observation>
                </component>
                <component>
                        <observation classCode="OBS" moodCode="EVN">
                                <templateId root="
                                    ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                                    ↪  Result observation template -->
                                <id root="c6f88322-67ad-11db-bd13-0800200
                                    ↪ c9a66"/>
                                <code code="27113001" codeSystem="
                                    ↪ 2.16.840.1.113883.6.96" displayName=
                                    ↪ "Body weight"/>
                                <statusCode code="completed"/>
                                <effectiveTime value="19991114"/>
```

```
                                <value codeSystem="2.16.840.1.113883.6.8"
                                    ↪ xsi:type="PQ" value="86" unit="kg"/>
                        </observation>
                </component>
                <component>
                        <observation classCode="OBS" moodCode="EVN">
                                <templateId root="
                                    ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                                    ↪  Result observation template -->
                                <id root="c6f88323-67ad-11db-bd13-0800200
                                    ↪ c9a66"/>
                                <code code="271649006" codeSystem="
                                    ↪ 2.16.840.1.113883.6.96" displayName=
                                    ↪ "Systolic BP"/>
                                <statusCode code="completed"/>
                                <effectiveTime value="19991114"/>
                                <value codeSystem="2.16.840.1.113883.6.8"
                                    ↪ xsi:type="PQ" value="132" unit="mm[
                                    ↪ Hg]"/>
                        </observation>
                </component>
                <component>
                        <observation classCode="OBS" moodCode="EVN">
                                <templateId root="
                                    ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                                    ↪  Result observation template -->
                                <id root="c6f88324-67ad-11db-bd13-0800200
                                    ↪ c9a66"/>
                                <code code="271650006" codeSystem="
                                    ↪ 2.16.840.1.113883.6.96" displayName=
                                    ↪ "Diastolic BP"/>
                                <statusCode code="completed"/>
                                <effectiveTime value="19991114"/>
                                <value codeSystem="2.16.840.1.113883.6.8"
                                    ↪ xsi:type="PQ" value="86" unit="mm[Hg
                                    ↪ ]"/>
                        </observation>
                </component>
        </organizer>
</entry>
<entry typeCode="DRIV">
        <organizer classCode="CLUSTER" moodCode="EVN">
                <templateId root="2.16.840.1.113883.10.20.1.35"/> <!--
                    ↪ Vital signs organizer template -->
                <id root="d11275e0-67ae-11db-bd13-0800200c9a66"/>
                <code code="46680005" codeSystem="2.16.840.1.113883.6.96"
                    ↪ displayName="Vital signs"/>
                <statusCode code="completed"/>
                <effectiveTime value="20000407"/>
```

```
<component>
        <observation classCode="OBS" moodCode="EVN">
                <templateId root="
                    ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                    ↪  Result observation template -->
                <id root="d11275e1-67ae-11db-bd13-0800200
                    ↪ c9a66"/>
                <code code="50373000" codeSystem="
                    ↪ 2.16.840.1.113883.6.96" displayName=
                    ↪ "Body height"/>
                <statusCode code="completed"/>
                <effectiveTime value="20000407"/>
                <value codeSystem="2.16.840.1.113883.6.8"
                    ↪ xsi:type="PQ" value="177" unit="cm"/
                    ↪ >
        </observation>
</component>
<component>
        <observation classCode="OBS" moodCode="EVN">
                <templateId root="
                    ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                    ↪  Result observation template -->
                <id root="d11275e2-67ae-11db-bd13-0800200
                    ↪ c9a66"/>
                <code code="27113001" codeSystem="
                    ↪ 2.16.840.1.113883.6.96" displayName=
                    ↪ "Body weight"/>
                <statusCode code="completed"/>
                <effectiveTime value="20000407"/>
                <value codeSystem="2.16.840.1.113883.6.8"
                    ↪ xsi:type="PQ" value="88" unit="kg"/>
        </observation>
</component>
<component>
        <observation classCode="OBS" moodCode="EVN">
                <templateId root="
                    ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                    ↪  Result observation template -->
                <id root="d11275e3-67ae-11db-bd13-0800200
                    ↪ c9a66"/>
                <code code="271649006" codeSystem="
                    ↪ 2.16.840.1.113883.6.96" displayName=
                    ↪ "Systolic BP"/>
                <statusCode code="completed"/>
                <effectiveTime value="20000407"/>
                <value codeSystem="2.16.840.1.113883.6.8"
                    ↪ xsi:type="PQ" value="145" unit="mm[
                    ↪ Hg]"/>
        </observation>
```

```
                        </component>
                        <component>
                                <observation classCode="OBS" moodCode="EVN">
                                        <templateId root="
                                            ↪ 2.16.840.1.113883.10.20.1.31"/> <!--
                                            ↪  Result observation template -->
                                        <id root="d11275e4-67ae-11db-bd13-0800200
                                            ↪ c9a66"/>
                                        <code code="271650006" codeSystem="
                                            ↪ 2.16.840.1.113883.6.96" displayName=
                                            ↪ "Diastolic BP"/>
                                        <statusCode code="completed"/>
                                        <effectiveTime value="20000407"/>
                                        <value codeSystem="2.16.840.1.113883.6.8"
                                            ↪ xsi:type="PQ" value="88" unit="mm[Hg
                                            ↪ ]"/>
                                </observation>
                        </component>
                </organizer>
        </entry>
</section>
</component>
</structuredBody>
</component>
</ClinicalDocument>
```

Listing D.3: HL7v2 ORUˆR01 sample, pipe-delimited (Health Level Seven International, 2009)

```
MSH|^~\&amp;|||||20000407130000+0500||ORUˆR01
PID|||996-756-495^^^2.16.840.1.113883.19.5||Levinˆ Henry|||M
OBR||||46680005
OBX|||50373000ˆBody height||177|cm|||||||19991114
OBX|||27113001ˆBody weight||86|kg|||||||19991114
OBX|||271649006ˆSystolic BP||132|mm[Hg]|||||||19991114
OBX|||271650006ˆDiastolic BP||86|mm[Hg]|||||||19991114|
```

Listing D.4: HL7v2 ORUˆR01 sample, XML (Health Level Seven International, 2009)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ORU_R01 xmlns="urn:hl7-org:v2xml">
        <MSH>
                <MSH.1>|</MSH.1>
                <MSH.2>^~\&amp;amp;</MSH.2>
                <MSH.7>
                        <TS.1>20000407130000+0500</TS.1>
                </MSH.7>
                <MSH.9>
                        <MSG.1>ORU</MSG.1>
                        <MSG.2>R01</MSG.2>
                </MSH.9>
```

```
</MSH>
<PATIENT_RESULT>
	<PATIENT>
		<PID>
			<PID.3>
				<CX.1>996-756-495</CX.1>
				<CX.4>
					<HD.1>2.16.840.1.113883.19.5</HD.1>
				</CX.4>
			</PID.3>
			<PID.5>
				<XPN.1>
					<FN.1>Levin</FN.1>
				</XPN.1>
				<XPN.2>Henry</XPN.2>
			</PID.5>
			<PID.8>M</PID.8>
		</PID>
	</PATIENT>
	<ORDER_OBSERVATION>
		<OBR>
			<OBR.4>
				<CE.1>46680005</CE.1>
			</OBR.4>
		</OBR>
		<G7O>
			<OBSERVATION>
				<OBX>
					<OBX.3>
						<CE.1>50373000</CE.1>
						<CE.2>Body height</CE.2>
					</OBX.3>
					<OBX.5>177</OBX.5>
					<OBX.6>
						<CE.1>cm</CE.1>
					</OBX.6>
					<OBX.14>
						<TS.1>19991114</TS.1>
					</OBX.14>
				</OBX>
			</OBSERVATION>
			<OBSERVATION>
				<OBX>
					<OBX.3>
						<CE.1>27113001</CE.1>
						<CE.2>Body weight</CE.2>
					</OBX.3>
					<OBX.5>86</OBX.5>
					<OBX.6>
```

```
                                        <CE.1>kg</CE.1>
                                    </OBX.6>
                                    <OBX.14>
                                        <TS.1>19991114</TS.1>
                                    </OBX.14>
                                </OBX>
                            </OBSERVATION>
                            <OBSERVATION>
                                <OBX>
                                    <OBX.3>
                                        <CE.1>271649006</CE.1>
                                        <CE.2>Systolic BP</CE.2>
                                    </OBX.3>
                                    <OBX.5>132</OBX.5>
                                    <OBX.6>
                                        <CE.1>mm[Hg]</CE.1>
                                    </OBX.6>
                                    <OBX.14>
                                        <TS.1>19991114</TS.1>
                                    </OBX.14>
                                </OBX>
                            </OBSERVATION>
                            <OBSERVATION>
                                <OBX>
                                    <OBX.3>
                                        <CE.1>271650006</CE.1>
                                        <CE.2>Diastolic BP</CE.2>
                                    </OBX.3>
                                    <OBX.5>86</OBX.5>
                                    <OBX.6>
                                        <CE.1>mm[Hg]</CE.1>
                                    </OBX.6>
                                    <OBX.14>
                                        <TS.1>19991114</TS.1>
                                    </OBX.14>
                                </OBX>
                            </OBSERVATION>
                        </G7O>
                    </ORDER_OBSERVATION>
                </PATIENT_RESULT>
</ORU_R01>
```

Appendix E

SAP for Healthcare

Figure E.1: Product overview of SAP ECC

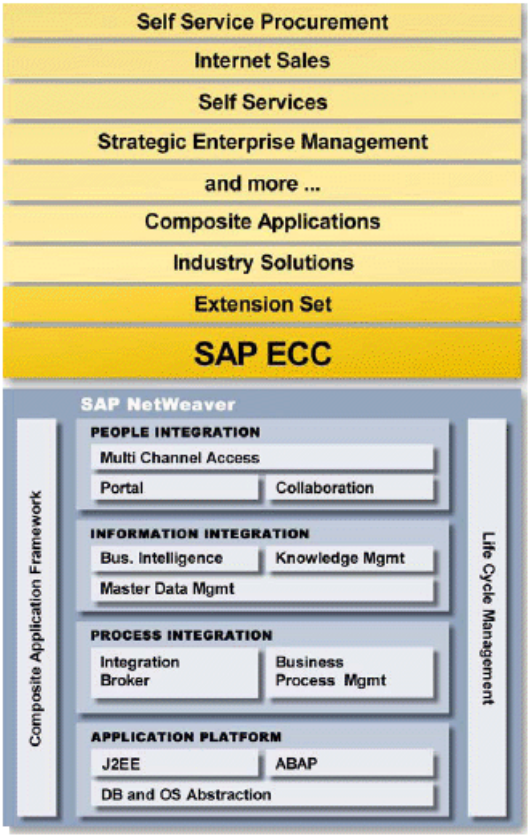


Figure E.2: Product overview of SAP for Healthcare (SAP SE, 2014b)

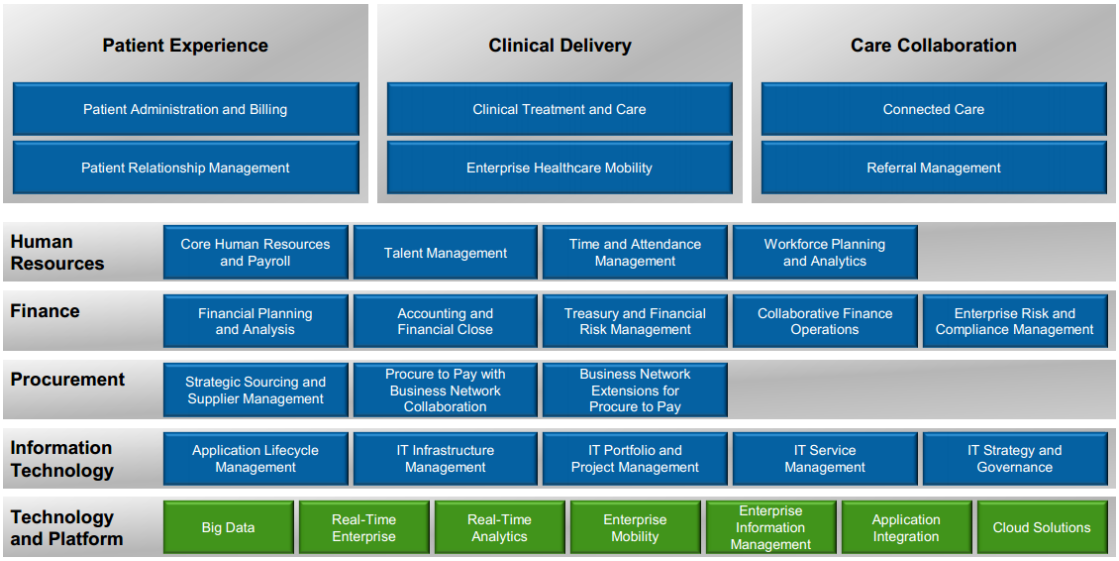
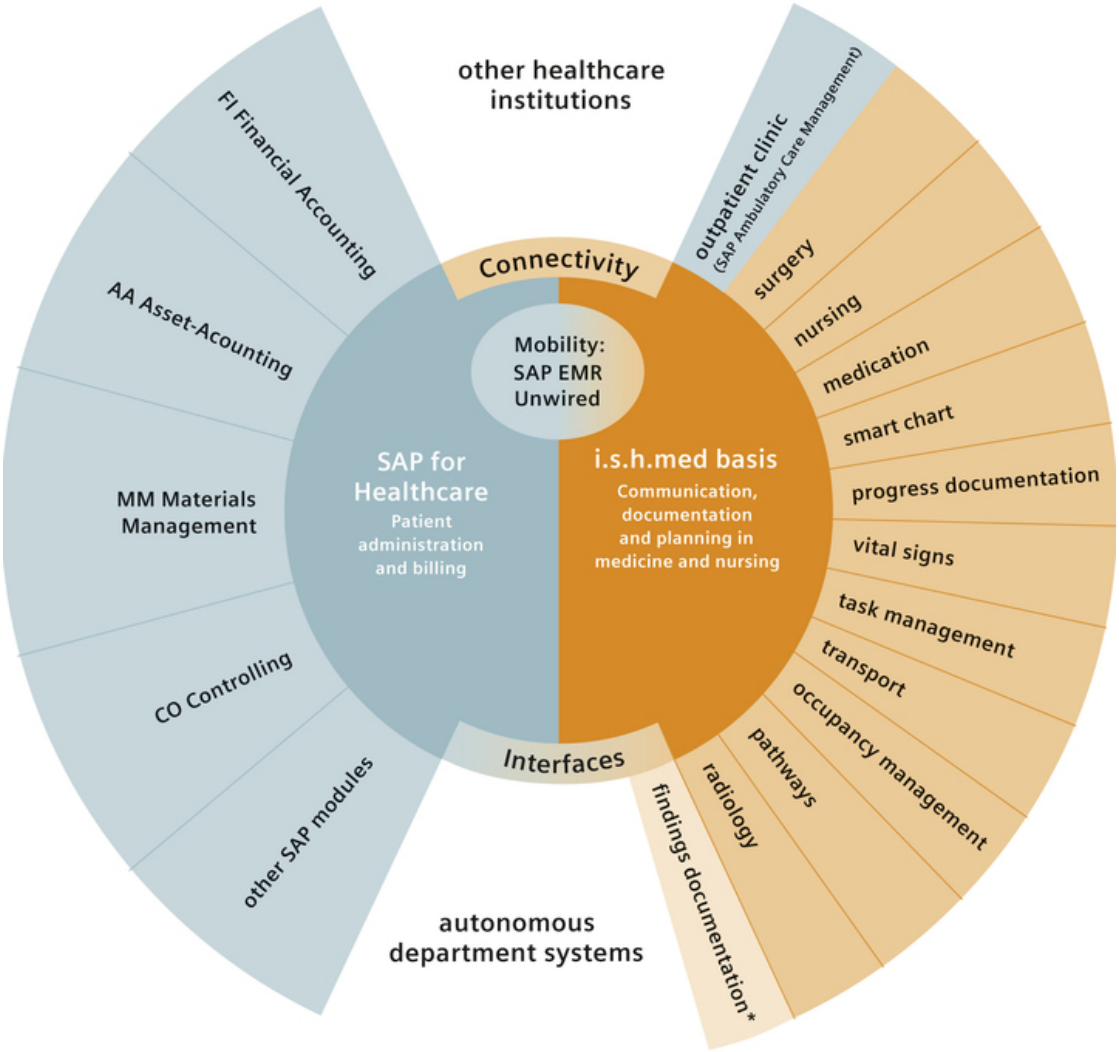


Figure E.3: Modules overview of SAP PAB and i.s.h.med (Siemens AG, 2014)



Appendix F

Microsoft HealthVault

Table F.1: Microsoft HealthVault API Methods (Microsoft, 2014c)

AddApplication	Adds an application configuration.
AllocatePackageId	The unique identity code (to access the package) or failure result.
AssociateAlternateId	Associates an alternate id with a person and record.
AuthorizeApplication	
BeginPutBlob	The authentication token to be supplied with a streaming put blob request.
BeginPutConnectPackageBlob	
BeginSignalDownload	Schema for the BeginSignalDownload method request.
BeginSignalUpload	Schema for the BeginSignalUpload method request.
CreateAuthenticatedSessionToken	Provides a way for clients to establish an authenticated session with Microsoft HealthVault.
CreateConnectPackage	Creates a package containing a data that the user can claim using the identity code returned by this method
CreateConnectRequest	Creates a connect request which will allow the user to tie their HealthVault account to the calling application once the user validates the connect request.
DeletePendingConnectPackage	Deletes the pending connect package.
DeletePendingConnectRequest	Deletes the pending connect request for the supplied external id.
DisassociateAlternateId	Disassociates an alternate id with a person and record.
GetAlternateIds	
GetApplicationInfo	Gets the settings for the current application.
GetApplicationSettings	Gets the application specific settings for the person.

Table F.1: Microsoft HealthVault API Methods (Microsoft, 2014c)

GetAuthorizedConnectRequests	Returns all found authorized connect requests associated with the calling application.
GetAuthorizedPeople	Gets information about the people that are authorized for an application.
GetAuthorizedRecords	Gets information about the specified health records for which the authenticated person is authorized to use with the calling application.
GetEventSubscriptions	The response containing the entire collection of subscriptions for calling application.
GetMeaningfulUseTimelyAccessReport	Gets the Meaningful Use Timely Access Report for an application.
GetMeaningfulUseVDTReport	Gets the Meaningful Use VDT Report for an application.
GetPeopleForRecord	Gets information about the people that have been authorized to a record or invited to share a record.
GetPersonAndRecordForAlternateId	Gets the person and record ids associated with an alternate id.
GetPersonInfo	Gets basic information about the authenticated person.
GetRecordOperations	Gets all operations that have occurred on the record since a specified sequence number.
GetServiceDefinition	This method is used to get information about Microsoft HealthVault and its related features.
GetSignalSources	Schema for the GetSignalSources method request.
GetThings	Searches for things based on supplied criteria.
GetThingType	Gets information about the thing types supported by the platform.
GetUpdatedRecordsForApplication	Gets a list of records for an application with things that have been updated since a specified date.
GetValidGroupMembership	Gets a list of valid things of type Group Membership.
GetVocabulary	Gets the code items associated with a vocabulary.
NewApplicationCreationInfo	Gets information needed to authorize a new SODA application instance.
NewSignupCode	The unique signup code or failure result.
OverwriteThings	Overwrites a thing even if the new data uses an older version of the thing schema.
PutThings	Creates or updates one or more things.
QueryPermissions	Gets the permissions the authenticated person has for the specified thing type for the specified record.

Table F.1: Microsoft HealthVault API Methods (Microsoft, 2014c)

RemoveApplicationRecordAuthorization	
RemoveThings	Deletes things from a record.
SearchVocabulary	Searches a vocabulary and retrieves code items that match a given search criteria.
SelectInstance	Gets the HealthVault instance that supports a specified geographic location.
SendInsecureMessage	Sends an insecure email message to the specified recipients.
SendInsecureMessageFromApplication	Sends an insecure email message originating from the application to the specified recipients.
SetApplicationSettings	Sets the application specific settings for the person.
SubscribeToEvent	Creates an event subscription for the calling application.
UnsubscribeToEvent	Removes the subscription identified by the supplied id.
UpdateApplication	Updates an application configuration.
UpdateEventSubscription	Updates an event subscription for the calling application.
UpdateExternalId	Updates the external id for a connect request.

Table F.2: Microsoft HealthVault Data Types (Microsoft, 2014a)

Advance directive	An advance directive such as a living will.
Aerobic profile	A summary of a person's aerobic condition.
Allergic episode	A single instance of an allergic reaction.
Allergy	A hypersensitivity to an allergen.
App-specific information	Arbitrary or custom data for use by an application.
Application data reference	Information that can be used by an application to render content from another application.
Appointment	A medical appointment.
Asthma inhaler	An inhaler unit used to treat asthma.
Asthma inhaler usage	A single use of an inhaler.
Basic demographic information	Defines a set of data about the health record that is considered not to be personally-identifiable.
Blood glucose	A single blood glucose reading.
Blood oxygen saturation	The percentage of oxygen saturation in the blood.
Blood pressure	A single blood pressure reading.
Body composition	A body composition measurement.
Body dimension	A body dimension such as waist size or head circumference.
Calorie guideline	A guideline for caloric intake.
Cardiac profile	A summary of a person's cardiac condition.
Care plan	A care plan containing tasks and goals.

Table F.2: Microsoft HealthVault Data Types (Microsoft, 2014a)

Cholesterol	A single cholesterol reading.
Clinical Document Architecture (CDA)	No summary available.
Comment	A comment associated with another data item.
Concern	A concern that a person has about a condition or life issue.
Condition	A medical issue or problem.
Contact	A contact such as an emergency contact, doctor, lawyer, etc.
Continuity of Care Document (CCD)	No summary available.
Continuity of Care Record (CCR)	No summary available.
Contraindication	A substance that interacts badly with a medical condition or drug.
Daily dietary intake	The amount of dietary nutrients and minerals consumed in a day.
Daily medication usage	A record of taking a medication or dietary supplement.
Defibrillator episode	The data from an implantable defibrillator after an episode.
Diabetic profile	A summary of a person's diabetic condition.
Discharge summary	A summary of a discharge from a health provider.
Education - MyData file (preview)	A MyData education file.
Education - SIF student academic record (preview)	No summary available.
Education document (preview)	An education document such as an assignment or exam.
Emotional state	A subjective record of an emotional state.
Encounter	A medical encounter such as an annual physical.
Exercise	An exercise session such as running or climbing.
Exercise samples	A series of data samples from an exercise session.
Explanation of benefits (EOB)	An explanation of benefits received from an insurance plan.
Family history	A condition of a relative.
Family history condition	A condition of a relative.
Family history person	Information about a relative of the record owner.
File	A file that can be uploaded to a health record in Microsoft HealthVault.
Food & drink	The amount of dietary nutrients and minerals consumed.
Genetic SNP result	A collection of results from a SNP genetic test.
Group membership	Memberships of the record owner.
Group membership activity	An activity related to group membership.
HbA1C	An HbA1C reading that measures the amount of glycosylated hemoglobin.

Table F.2: Microsoft HealthVault Data Types (Microsoft, 2014a)

Health assessment	The results of a health assessment such as a diabetes assessment.
Health event	A general health event such as the first time a baby crawls.
Health goal	A health goal that defines a target goal such as steps per day.
Health journal entry	An entry of a health journal or diary.
Healthcare proxy	A healthcare proxy that appoints an agent to make medical decisions.
Heart rate	A heart rate measurement in beats per minute.
Height	A single height measurement.
Immunization	An immunization to prevent a disease or condition.
Insulin injection	An insulin injection used to treat diabetes.
Insulin injection usage	A single use of an insulin injection.
Insurance plan	A person or organization that pays for health and medical bills.
Lab results	A series of lab test results.
Life goal	A general life goal such as to travel or quit smoking.
Meal definition	A meal that is commonly eaten or a meal associated with a particular diet plan.
Medical annotation	A medical annotation containing transcribed notes and other information.
Medical device	A piece of medical equipment such as a blood pressure reader or pedometer.
Medical image study	A study containing medical images.
Medical problem	A medical problem and diagnosis.
Medication	A substance used for the treatment of a disease or condition.
Medication fill	Information related to filling a medication.
Menstruation	A single assessment of menstrual flow.
Message	A multipart message including message text and attachments.
Microbiology lab test result	A microbiology lab test result.
PAP session	A Positive Airway Pressure (PAP) session.
Password-protected package	A package that contains a pkcs5v2 encrypted blob.
Peak flow	A peak flow measurement used to track lung function.
Personal contact information	The contact information for the person owning a health record in Microsoft HealthVault.
Personal demographic information	Personal demographic information that is considered sensitive in nature.
Personal picture	An image that represents the person.
Pregnancy	A record of a pregnancy and delivery.
Procedure	A medical procedure and results.
Question & answer	A question that was asked and the answers given.
Radiology result	The results of a radiology lab test.
Respiratory profile	A summary of a person's respiratory condition.
Sleep journal entry	A daily journal of activities that impact sleep.

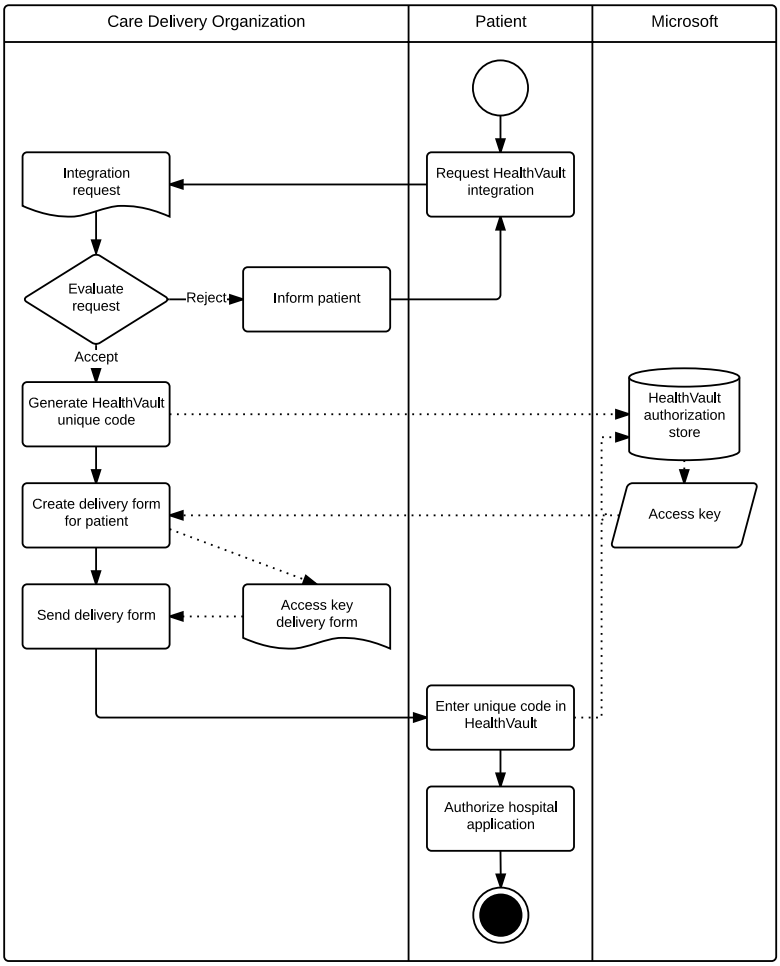
Table F.2: Microsoft HealthVault Data Types (Microsoft, 2014a)

Sleep session	A sleep journal entry made in the morning to reflect on the prior night.
Status	The status of an item in the health record.
Vital signs	A set of vital signs such as body temperature.
Web link	A link to a web page.
Weekly aerobic exercise goal	A weekly goal for aerobic exercise.
Weight	A single weight measurement.
Weight goal	A target weight range with an associated target date.

Appendix G

Prototype

Figure G.1: Patient registration process



Listing G.1: HL7 ORUˆR01 2.5.1 example message (XML)

```
<?xml version='1.0' encoding='utf-8'?>
<ORU_R01 xmlns='urn:hl7-org:v2xml'>
       <MSH xmlns='urn:hl7-org:v2xml'>
              <MSH.1>|</MSH.1>
              <MSH.2>^~\&amp;</MSH.2>
              <MSH.3>
                     <HD.1>CDO-SVEN</HD.1>
              </MSH.3>
              <MSH.4>
                     <HD.1>Sven's Academic Hospital</HD.1>
              </MSH.4>
              <MSH.5>
                     <HD.1>HV</HD.1>
              </MSH.5>
              <MSH.6>
                     <HD.1>Microsoft HealthVault EU PPE</HD.1>
              </MSH.6>
              <MSH.7>
                     <TS.1>201502171258</TS.1>
              </MSH.7>
              <MSH.9>
                     <MSG.1>ORU</MSG.1>
                     <MSG.2>R01</MSG.2>
                     <MSG.3>ORU_R01</MSG.3>
              </MSH.9>
              <MSH.10>13446756 </MSH.10>
              <MSH.11>
                     <PT.1>P</PT.1>
              </MSH.11>
              <MSH.12>
                     <VID.1>2.5.1</VID.1>
              </MSH.12>
              <MSH.16>NE</MSH.16>
              <MSH.21 xmlns='urn:hl7-org:v2xml'>
                     <EI.1>hwrProfile</EI.1>
                     <EI.3>2.16.840.1.113883.9.29</EI.3>
                     <EI.4>ISO</EI.4>
              </MSH.21>
       </MSH>
       <ORU_R01.PATIENT_RESULT xmlns='urn:hl7-org:v2xml'>
              <ORU_R01.PATIENT xmlns='urn:hl7-org:v2xml'>
                     <PID>
                            <PID.3>
                                   <CX.1>0000033467</CX.1>
                                   <CX.4>
                                          <HD.1>CDO-SVEN</HD.1>
                                   </CX.4>
                            </PID.3>
```

```
                <PID.5>
                        <XPN.1>
                                <FN.1>SVEN</FN.1>
                        </XPN.1>
                        <XPN.2>BASTIANEN</XPN.2>
                        <XPN.7>L</XPN.7>
                </PID.5>
                <PID.7>
                        <TS.1>19900310</TS.1>
                </PID.7>
                <PID.8>M</PID.8>
        </PID>
</ORU_R01.PATIENT>
<ORU_R01.ORDER_OBSERVATION>
        <OBR xmlns='urn:hl7-org:v2xml'>
                <OBR.1>1</OBR.1>
                <OBR.3>
                        <EI.1>0000000060</EI.1>
                </OBR.3>
                <OBR.4>
                        <CE.1>HWR</CE.1>
                        <CE.2>HWR</CE.2>
                        <CE.3>Height and weight report</CE.3>
                        <CE.4>L</CE.4>
                </OBR.4>
                <OBR.7>
                        <TS.1>201502171258</TS.1>
                </OBR.7>
                <OBR.22>
                        <TS.1>201502171257</TS.1>
                </OBR.22>
                <OBR.25>F</OBR.25>
        </OBR>
        <ORU_R01.OBSERVATION>
                <OBX xmlns='urn:hl7-org:v2xml'>
                        <OBX.1>1</OBX.1>
                        <OBX.2>NM</OBX.2>
                        <OBX.3>
                                <CE.1>3141-9</CE.1>
                                <CE.2>Body weight measured</CE.2>
                                <CE.3>LN</CE.3>
                        </OBX.3>
                        <OBX.5>78.5</OBX.5>
                        <OBX.6>
                                <CE.1>kg</CE.1>
                                <CE.2>kilogram</CE.2>
                                <CE.3>UCUM</CE.3>
                        </OBX.6>
                        <OBX.11>F</OBX.11>
```

```
                                    <OBX.14>
                                        <TS.1>201502171258</TS.1>
                                    </OBX.14>
                                </OBX>
                            </ORU_R01.OBSERVATION>
                        </ORU_R01.ORDER_OBSERVATION>
                    </ORU_R01.PATIENT_RESULT>
</ORU_R01>
```

Table G.1: Sequence diagrams: Systems and components

adapter	The core module of the HealthVault Integrator that handles all communication with the HealthVault platform.
clerk	A clerk of a CDO.
hl7Parser	A module of the HealthVault Integrator that is capable of parsing and encoding HL7v2 messages.
is2	The i.s.h.med instance that is used in the prototype setup.
oru_R01_SAP_In	The inbound ABAP proxy, used when a new weight measurement is received for i.s.h.med.
oru_R01_SAP_Out	The outbound ABAP proxy, used when a new weight measurement is stored in i.s.h.med.
patientData	A file system folder in the LAN of a CDO where a small data set of registered patients is stored.
patientManager	A module of the HealthVault Integrator that contains the web interface that is used to register a patient.
po4	The SAP PI instance that is used in the prototype setup.
pre-production	The pre-production instance of Microsoft HealthVault that is used in the prototype.
watchFolder	A file system folder in the LAN of a CDO where intermediate HL7 messages are temporarily stored for subsequent processing.

Installation guide

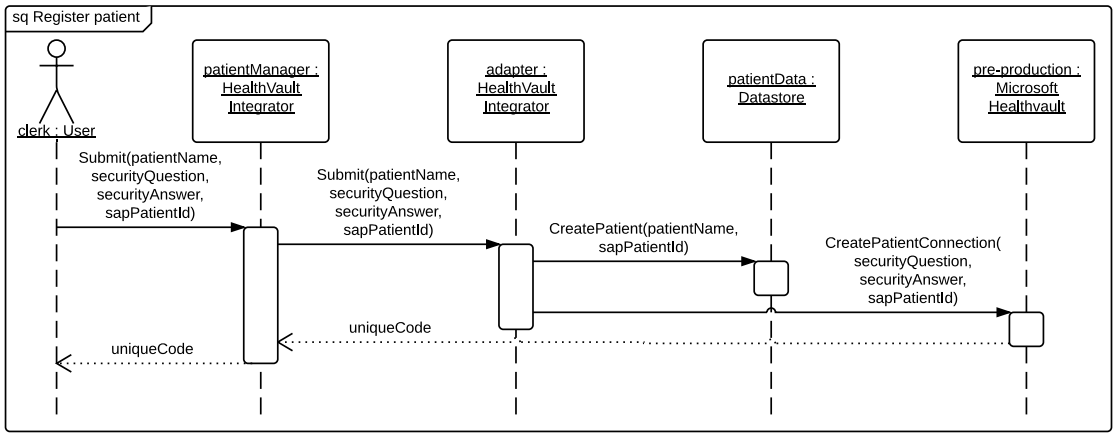
In this section the steps for deploying this prototype at some CDO are broadly described. Due to the fact that an free-standing, intermediate application, i.e. the HealthVault Integrator, has been built, the prototype can be deployed with limited effort. The prerequisites for this prototype are:

- An instance of ECC including PAB and i.s.h.med.
- An instance of SAP PI.
- An internet-enabled server with a Windows operating system.
- A folder on the file system that the SAP PI instance and the Windows server can access.

Subsequently the following steps have to be executed in order to deploy the prototype:

1. HealthVault Integrator

Figure G.2: Sequence diagram: Register patient



a) Register an application at the HealthVault platform and download the related certificate and application ID.

b) Install the HealthVault Integrator and assign the certificate and application ID.

c) Configure the directories for the watch folder and the patient data folder.

2. SAP PI

a) Register the HealthVault Integrator as a *Business System* in the SAP System Landscape Directory (SAP SLD), such it gets recognized by SAP systems.

b) Load the XSD of the message HL7v2.5.1 ORUˆR01.

c) Import the *Design object* into the *Enterprise Services Builder* of SAP PI.

d) Import the *Integration Flows* of this prototype and assign them to the input and output interfaces.

e) Configure the directories for the watch folder and the patient data folder in the Integration Flows.

3. SAP for Healthcare

a) Create the organizational unit *Patient*, such that it can be indicated that the creator of a vital sign measurement is the patient.

b) Generate the ABAP proxies.

c) Import the code into the inbound ABAP proxy.

d) Import the code for the custom function module that calls the outbound ABAP proxy.

e) Import the code of the BAdI.

f) Configure the correct vital sign type that is used for weight measurements.

On the condition that the prerequisites are met, the installation of the prototype will take approximately one day. This implies

Figure G.3: Sequence diagram: Create HL7 message from i.s.h.med observation

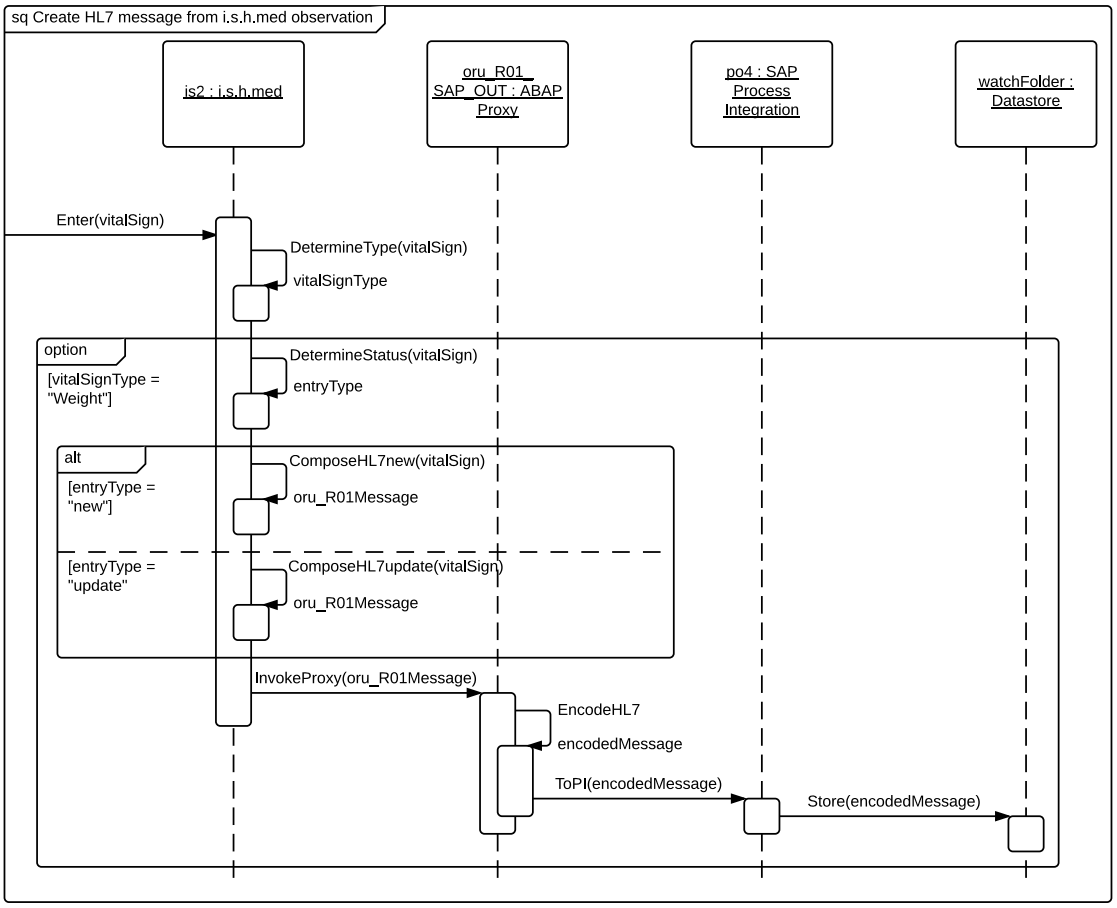


Figure G.4: Sequence diagram: Send HL7 message to HealthVault

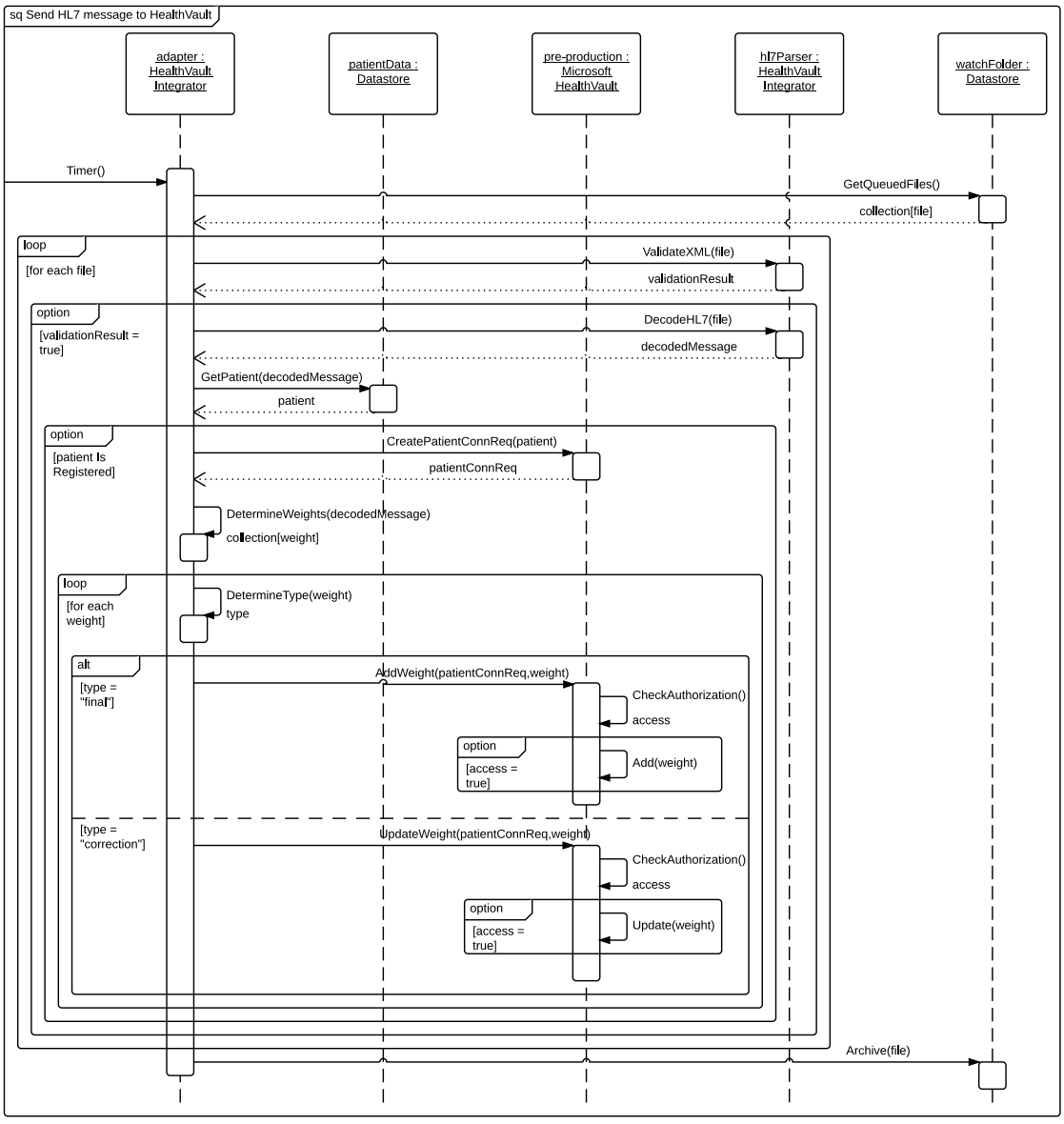


Figure G.5: Sequence diagram: Create HL7 message from HealthVault observation

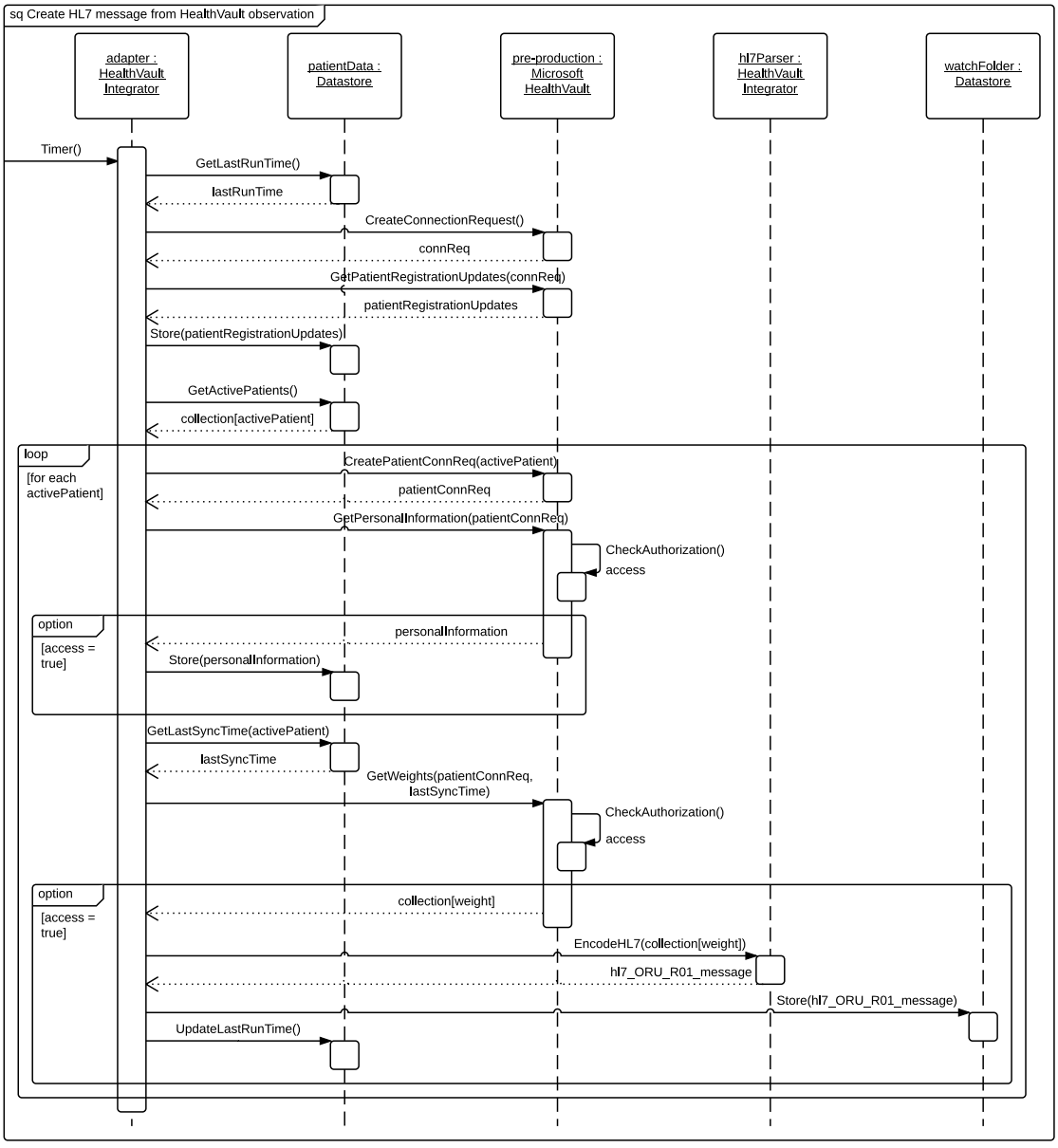
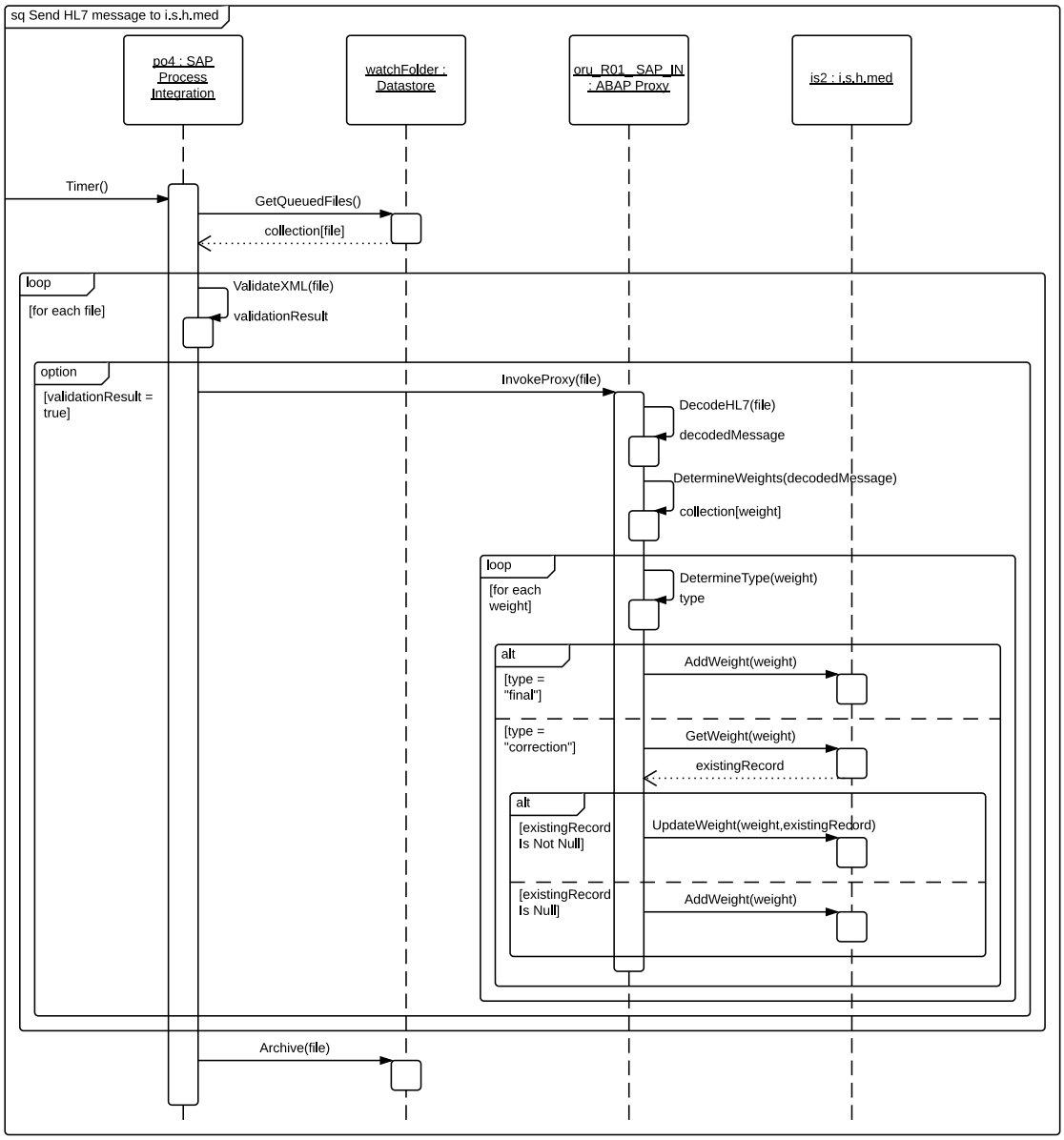


Figure G.6: Sequence diagram: Send HL7 message to i.s.h.med



Appendix H

Interview minutes

This appendix contains the minutes of the formal interviews that have been held for this research.

Table H.1: Interview Robert Duivenvoorden (October 14, 2014)

Interviewer	Sven Bastianen
Interviewee	Robert Duivenvoorden
Interviewee role	Managing consultant at SAP NL
Date	October 14, 2014. 11:00 AM - 11:45 AM
Location	SAP Nederland B.V., 's-Hertogenbosch
Topic	Interoperability options for SAP ECC
Minutes validated?	Yes

SAP offers a wide range of options to provide interoperability possibilities or integration platforms for its products. Depending on the required functionality and the system landscape a choice for one of these options can be made. Due to business and technological reasons some interoperability options are obsolete nowadays, but are still supported because of the large install base that these products have. This implies that these options are not actively promoted or sold anymore, but can still be installed by customers of sap. The following techniques can be used in integration scenarios.

ABAP Proxy An ABAP Proxy is an interface in ECC that can send or receive a message. It can be called by an external application with SOAP and communicates over the HTTP. The big advantage of an ABAP Proxy is that it can trigger an ABAP program to run in ECC natively, while it is invoked externally. Typically an ABAP Proxy is generated based on an XSD provided by SAP PI.

SAP Business Connector (SAP BC) SAP BC is middleware software that can be used on top of ECC in order to integrate ECC and external (non-SAP) systems. It supports various protocols, i.e. HTTP and FTP, and communicates to an ECC system with RFC. SAP BC is reaching the end of it lifecycle, as the product SAP PI is the successor of SAP BC.

SAP Connector for Microsoft .NET (SAP NCo) SAP Connector for Microsoft .NET (SAP NCo) is a SDK for the .NET framework. This kit enables developers to integrate a .NET

application with an SAP system. The connector can manage the communication between these two systems with SOAP and a RFC.

SAP Java Connector (SAP JCo) SAP Java Connector (SAP JCo) is an SDK for Java. This kit enables developers to integrate a Java applications with an SAP system. These applications are able call remote functions that are defined in an SAP system.

SAP NetWeaver RFC Library SAP NetWeaver RFC Library is a library that can be used in applications that are written in C and C++ programming languages. It enables these applications to use RFC in order to communicate to an ECC system.

SAP Process Integration (SAP PI) SAP PI is the integration platform of SAP. SAP PI acts as a so-called *integration broker* or *message-oriented middleware*, which means that it is capable of routing, queueing, and transforming messages. It can be seen as a standalone application that supports various protocols, standards and the mapping between the various interfaces (so-called *adapters*). It is developed for integrating SAP and non-SAP systems.

Remote Function Call (RFC) RFC is an SAP interface that enables external systems to invoke a function within an ECC system. The functionality of RFC is not an integration technique on itself, but is often used within one of the integration methods to set up communication to an ECC system, i.e. SAP BC.

In the situation that an external system has to interoperate with an SAP system, the most preferable solution is to use SAP PI in combination with ABAP proxies.

Table H.2: Interview Ronald de Zoete (November 25, 2014)

Interviewer	Sven Bastianen
Interviewee	Ronald de Zoete
Interviewee role	Product Manager i.s.h.med at Siemens NL / Cerner
Date	November 25, 2014. 2:00 PM - 4:00 PM
Location	Siemens Nederland, 's-Gravenhage
Topic	i.s.h.med presentation and support
Minutes validated?	Yes

i.s.h.med is the clinical extension of SAP IS–H. It provides functionality for physicians, nurses and other staff of a healthcare provider to document and store diagnoses, medical findings, lab results etcetera. i.s.h.med is built in a separate namespace within the software of SAP ECC. This means that although i.s.h.med is a separate product formally, it is integrated with SAP IS–H to such an extent that a user cannot distuingish SAP IS–H and i.s.h.med while using the software. In terms of data structures, i.s.h.med reuses relevant parts of SAP IS–H and provides default structures for regular data types like lab results. Every i.s.h.med-using party has these default structures. Next to this, i.s.h.med offers the functionality of Parameterized Medical Documentation (PMD), which enables the user to design and implement custom forms. Underlying programs, data structures, data types etcetera are automatically created by the system. PMD may also rely on existing functionality or structures and could for example be used for entering lab results, which is subsequently stored in the default data structure of i.s.h.med.

Weight measurements When considering the case of a weight measurement, this is ideally implemented in i.s.h.med as a *vital sign*. Vital signs are part of the default structure of i.s.h.med and is present in every i.s.h.med system. Siemens has included various BAdIs in i.s.h.med. A BAdI is a general concept within ECC that enable end users to implement custom code that gets called by standard events, i.e. when a vital sign is saved. In case we want to add functionality to i.s.h.med at the event that a vital sign is created, it is recommended to make use of the BAdI that has been implemented for this purpose.

Table H.3: E-mail conversation Alf Zwilling (January 30, 2015)

Interviewer	Sven Bastianen
Interviewee	Alf Zwilling
Interviewee role	Manager Corporate Communication at VZVZ
Date	January 30, 2015
Topic	Q&A about LSP

Which (type of) consents can be currently defined by a patient in LSP? Is this limited to the options offered at the website `www.ikgeeftoestemming.nl`**?** Yes. A patient can see which CDOs have registered data about him at LSP and the patient can see which CDOs have requested data about him: When and what type of data.

In case a CDO has requested data at LSP based on the provided consent of a patient. In case a patient withdraws his consent: What are the rights and obligations does this CDO with respect the earlier retrieved data? The requesting CDO will not store any retrieved data in his EMR system. However, he will send a message to the general practitioner of the patient describing the care that he has delivered to the patient.

A CDO can register patient data at LSP. How does dit exactly look like? Instinctively I would say that there is predefined list of data types that can be registered at LSP. Is this done via a standard provided by HL7 or is this implemented differently? The answer is twofold:

1. Standards are defined describing to which format a message should comply. This is according to HL7.
2. The content is defined as well; The *Professionele Samenvatting (PS)*[1] contains recent information from the general practitioner conforming to the guidelines of the profession. The medication overview is composed based on the list of supplied medication to the patient, also conforming to the guidelines set by pharmacists.

In the meeting of the NICTIZ/VZVZ-focus group meeting in October 2014, the topic *The patient in the role of requesting party at LSP* was discussed. Is there any news on this topic? Yes. This answer is twofold as well:

1. In the proposed amendment of *Wet Cliëntrechten in de zorg*[2] this new role of the patient is covered.
2. Regarding the development of a PHR, this is an option that is seriously considered.

---

[1] Dutch term for "Professional summary".
[2] Dutch law name saying "Law for Client rights in healthcare".